CIO-2061

National Imagery & Mapping Agency

Common Imagery Interoperability Facilities Reference Model

Version 2.0
20 December 1996

PREFACE

This document was prepared by Booz·Allen & Hamilton Inc. for the National Imagery and Mapping Agency. Comments or requests for additional information are welcome and should be addressed to:

National Imagery and Mapping Agency
Attn: SEIT
14675 Lee Road
Chantilly, VA 20151-1715

Additional copies of this or related documents may be obtained by written request.

TBD/TBR/TBS LISTING

Section	*Pages*	*Control Number*	*Description*
4.2.2.4	29	TBR001	The definition of the Geolocation Facility may require additional partitioning or other refinements.
4.2.2.6	29	TBR002	The definition of the Automatic Target Recognition domain interface may require additional partitioning or other refinements.
4.2.2.7	30	TBR003	The definition of the Image Synthesis domain interface may require additional partitioning or other refinements.
4.2.2.8	31	TBR004	The definition of the Image Understanding domain interface may require additional partitioning or other refinements.
4.2.3	32	TBR005	The definition of the Requirements Management Support Services may require additional partitioning or other refinements.
4.2.3	32	TBR006	The definition of the Exploitation Management Support Services may require additional partitioning or other refinements.
4.2.3	32	TBR007	The definition of the Dissemination Management Support Services may require additional partitioning or other refinements.
4.2.4	32	TBR008	The Mapping, Charting, and Geodesy Domain Interfaces are yet to be defined.
4.3	33	TBR009	The definition of the Imagery Compression Facility may require additional partitioning or other refinements.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Title** **Page**

LIST OF TABLES

Section 1
INTRODUCTION

1.1 Purpose

The *Common Imagery Interoperability Facilities Reference Model* specifies a framework for developing an open application program interface (API) between architectural elements of the United States Imagery and Geospatial Information System (USIGS). This framework will serve both as a technical specification for the eventual development of the individual API definitions, and as a management tool for planning and controlling the work required to develop these facilities. Additional insights into the management and execution of this work is contained in the *Common Imagery Interoperability Working Group Management Plan*.

The interface and facility requirements documented within address objective needs of the USIGS. The realities of time, budget, and available technology limitations are expected to constrain initial CIIF implementations. Programmatic details pertaining to schedules and scope of implementation initiatives will be captured in other documentation.

Although the primary motivation for defining the CIIF is to support the development of the USIGS, these interface standards are desired and expected to be applicable to other application domains.

1.2 Scope

The USIGS Technical Architecture is defined in terms of a collection of “elements,” each of which comprises a collection of distinct services. Together, these elements and services describe the end-to-end imagery cycle. The element services are defined in the *USIS Technical Architecture Requirements* document by specifying their functional, performance, and interface requirements, and by referencing applicable *USIS Standards & Guidelines* paragraphs. These standards address not only communications protocols and data formats, but also the important issue of service-to-service interactions. The technical architecture of these service-to-service interactions is the essential focus of the present document.

Although the USIGS Technical Architecture encompasses nearly all the transaction types that occur within the imagery business in general, this reference model is more narrowly focused on services provided inside the boundaries of the USIGS, and even more specifically on those interfaces that require standardization within the USIGS. The interfaces themselves represent device and location independent software-to-software transactions. This document identifies interfaces that address related API functions, and groups them into interface architecture building blocks called “facilities.”

The interface architecture presented in this document is based on an object-oriented, distributed computing model that is similar in concept to emerging commercial standards, such as OMG’s Object Management Architecture. The CIIF Reference Model is comprised of four major architectural components:

- **Distributed Computing Infrastructure** – Defines the architecture and primitive interfaces for the communication mechanism that lies at the heart of the CIIF Reference Model
- **Object Service Interfaces Architecture** – Identifies the categories of fundamental service interfaces associated with the Distributed Computing Infrastructure
- **Common Facilities Architecture** – Describes certain higher-level service interfaces that can be used by many different kinds of applications
- **Geospatial Information Services** – Consists of standard interfaces that are uniquely tailored for the imagery and map-handling application domain.

1.3 Applicability

This document will determine the scope of (i.e., allocate functional requirements to) each of the Common Imagery Interoperability Facilities. These scope definitions will eventually guide the development of Interface Definition Language (IDL) for each facility. The finished IDL specifications will be incorporated into the *USIS Standards & Guidelines*. In addition, these specifications will be forwarded, as appropriate, to commercial standards groups, such as the OMG. It is anticipated that the finished IDL specifications will be used in the construction of USIGS applications.

1.4 Document Structure

- **Section 1: Introduction** – Provides an overview of the purpose, scope, applicability, and structure of the document.
- **Section 2: Applicable Documents** – Identifies other documentation that is either applicable to or referenced by this document.
- **Section 3: Interoperable Architecture Concepts** – Describes the high-level approach to distributed computing upon which the CIIF is based; also describes certain goals, conventions, and guidelines that should be applied to the process of deriving, defining, and developing a well-designed facility or imagery-specific interface.
- **Section 4: The CIIF Architecture** – Describes the overall structure and organization of the inter-related collection of interfaces and facilities that comprise the CIIF.
- **Appendices:** Definition of terms and acronyms.

Section 2
APPLICABLE DOCUMENTS

2.1 Referenced Documents

The following documents are cited within and form a part of this document to the extent specified herein:

- *USIS Standards & Guidelines* (CIO-2008), 13 October 1995
- *USIS Standards Profile for Imagery Archives*, Version 1.0, 20 July 1994
- *USIS Technical Architecture Requirements* (CIO-2004), 8 December 1995
- *Accelerated Architecture Acquisition Initiative (A3I) Requirements Document (ARD)*, CIO-2054, Rev. 2, 7 May 1996
- *Image Access Services Specification (IAS)*, CIO, Version 1.0, 24 May 1996
- *International Standards Organization (ISO) Draft International Standard 14750, Interface Definition Language (IDL)*, 1996.

2.2 USIGS-Related Documents

These documents provide additional information which may facilitate a reader's understanding of the material contained within this volume:

- *Accelerated Architecture Acquisition Initiative Joint Requirements Document Between The Central Imagery Office and The Defense Airborne Reconnaissance Office*,
  CIO-2044, 18 January 1996
- *The Central Imagery Office (CIO) and IMINT Directorate Joint Requirements Document for the United States Imagery System (USIS) 2000 Accelerated Architecture Acquisition Initiative (A3I) Requirements Document*, CIO-2042, Version 1.1,
  29 August 1996
- *Common Imagery Interoperability Working Group Management Plan*, 8 May 1996
- *Common Imagery Ground/Surface System Acquisition Standards Handbook (CIIGS-Hdbk)*, Defense Airborne Reconnaissance Office, Version 1.0, 9 June 1995
- *Defense Information Infrastructure Master Plan*, Version 4.0, 26 April 1996

- *Defense Information Infrastructure (DII) Common Operating Environment (COE) Integration and Runtime Specification (I&RTS)*, Version 2.0, 23 October 1995, updated 29 February 1996

- *Department of Defense Joint Technical Architecture*, Version 1.0, 9 July 1996

- *United States Imagery System Joint Requirements Document Between The Central Imagery Office and Rome Laboratory*, May 22, 1995

- *USIS Glossary* (CIO-2006)

- *USIS Objective Architecture Definition and Evolution* (CIO-2003), 8 December 1995.

2.3 Commercial Documents

- *The Component Object Model Specification*, Microsoft Corporation, 24 October 1994

- *OLE Programmer's Reference*--Volume I, Interfaces, API's, Enumerations, and Structures, Microsoft Corporation

- *OMG Object Management Architecture Guide (OMA Guide)*, Revision 2.0, Object Management Group, 1 September 1992, OMG TC Document 92.11.1

- *The OSF Distributed Computing Environment (DCE)*, Open Software Foundation.

Section 3
INTEROPERABLE ARCHITECTURE CONCEPTS

3.1 Distributed Computing Architecture Framework

This section introduces and describes the Technical Reference Model. The role of each of the model's architectural components is defined and briefly described. Particular emphasis is given to explaining the concept of Geospatial Information Services and Common Facilities, since these architectural components are the ones that will be adapted to fulfill most of the special interoperability needs of the USIGS. The remainder of this section describes key architectural goals, conventions, guidelines, and design considerations that have been adhered to during the development of the Technical Reference Model.

3.1.1 The Technical Reference Model

The Technical Reference Model, shown in Figure 3-1, intended to show how the USIGS architecture in general, and CIIF API standards in particular, relate to emerging architectural concepts and standards connected with the Defense Information Infrastructure Common Operating Environment (DII/COE). This reference model was derived by combining selected features of the:

- Current DII/COE reference model
- Object Management Group's Object Management Architecture (OMA) reference model
- Intelligence Community's draft reference model.

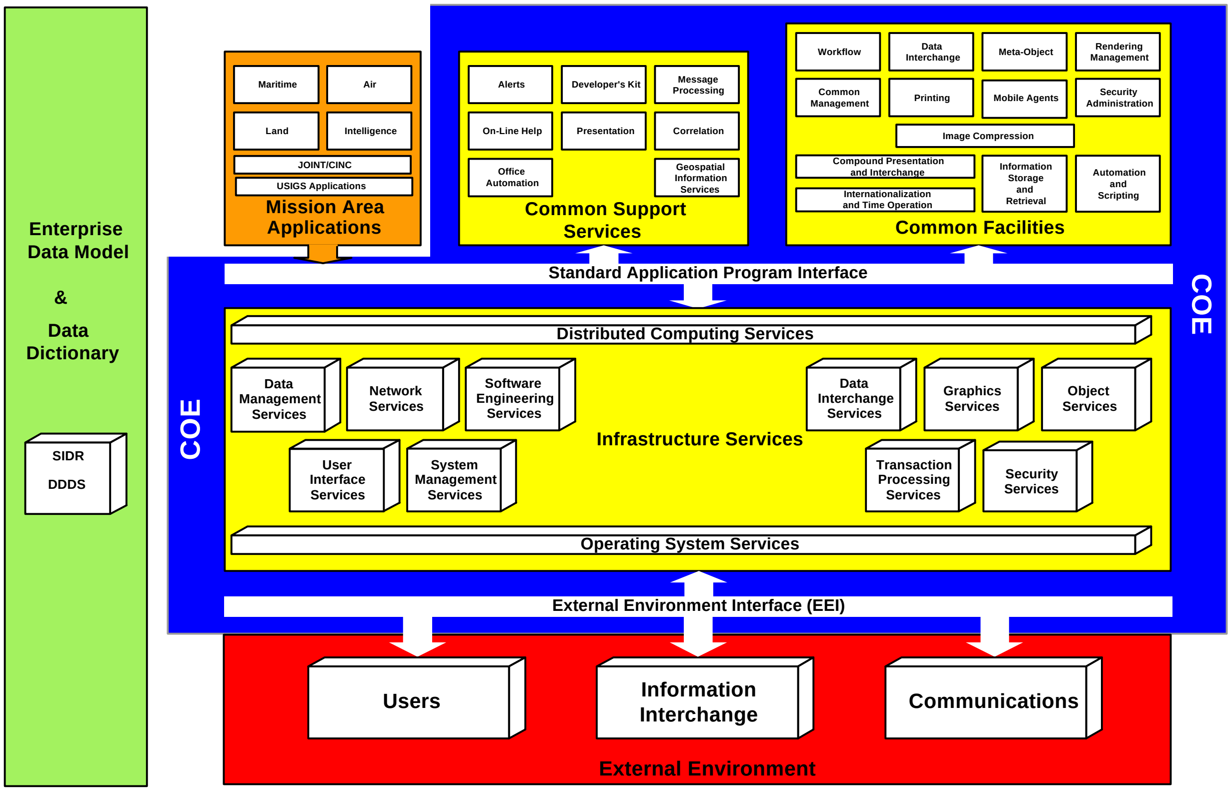

Figure 3-1 Technical Reference Model

This composite model lays the groundwork for migrating to a fully object-oriented distributing computing architecture, while recognizing that object-oriented and non-object based implementations will need to co-exist for some time to come. The key concepts contained in this model include:

- Use of Standard Application Program Interfaces to provide a fundamental enabling mechanism for better integrating the diverse collection of client applications, server applications, and operating system services that comprise the USIGS environment

- Promulgation of an open architecture that will use these standard APIs to facilitate the exchange of information and sharing of services among all manner of Mission Area Applications

- Establishment of a fundamental collection of Infrastructure Services (accessible via a standard API), which provide the basic capabilities needed to implement a distributed computing system architecture

- Definition of a collection of Common Support Services and Common Facilities, which provide a variety of ancillary services and utilities that are useful (and at time quite necessary) for building and standardizing distributed systems.

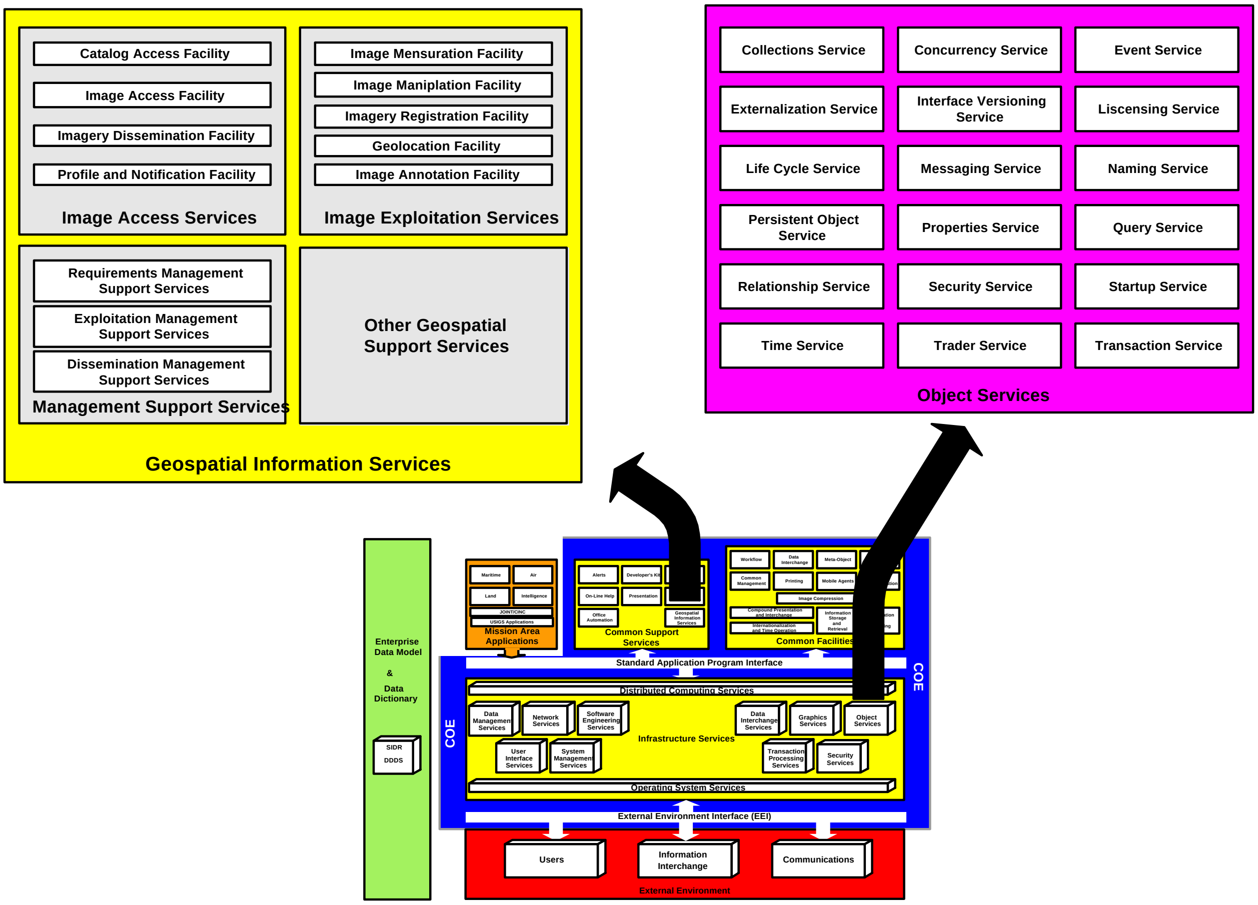
Catalog Access Facility
Image Access Facility
Imagery Dissemination Facility
Profile and Notification Facility
Image Access Services
Image Mensuration Facility
Image Maniplation Facility
Imagery Registration Facility
Geolocation Facility
Image Annotation Facility
Image Exploitation Services
Requirements Management Support Services
Exploitation Management Support Services
Dissemination Management Support Services
Management Support Services
Other Geospatial Support Services
Geospatial Information Services
Collections Service
Concurrency Service
Event Service
Externalization Service
Interface Versioning Service
Liscensing Service
Life Cycle Service
Messaging Service
Naming Service
Persistent Object Service
Properties Service
Query Service
Relationship Service
Security Service
Startup Service
Time Service
Trader Service
Transaction Service
Object Services
Enterprise Data Model
&
Data Dictionary
SIDR
DDDS
Maritime
Air
Land
Intelligence
JOINT/CINC
USIGS Applications
Mission Area Applications
Alerts
Developer's Kit
On-Line Help
Presentation
Office Automation
Geospatial Information Services
Common Support Services
Workflow
Data Interchange
Meta-Object
Common Management
Printing
Mobile Agents
Image Compression
Compound Presentation and Interchange
Internationalization and Time Operation
Information Storage and Retrieval
Common Facilities
Standard Application Program Interface
COE
Distributed Computing Services
Data Management Services
Network Services
Software Engineering Services
Data Interchange Services
Graphics Services
Object Services
Infrastructure Services
User Interface Services
System Management Services
Transaction Processing Services
Security Services
Operating System Services
External Environment Interface (EEI)
Users
Information Interchange
Communications
External Environment

Figure 3-2 CIIF Facilities and Object Service Interfaces

Two of the components included in the Technical Reference Model are particularly important in regard to understanding the connection between this model and the USIGS Software Reference Model (see Figure 3-3), as well as between this model and the details of the OMG OMA architecture. First, the Geospatial Information Services component of the Common Support Services contains the domain interface standards that are the primary focus of the CIIF. Second, the Object Services component of the Infrastructure Services contains the Object Service Interfaces that are fundamental to the OMG OMA architecture. These relationships are depicted Figure 3-2.

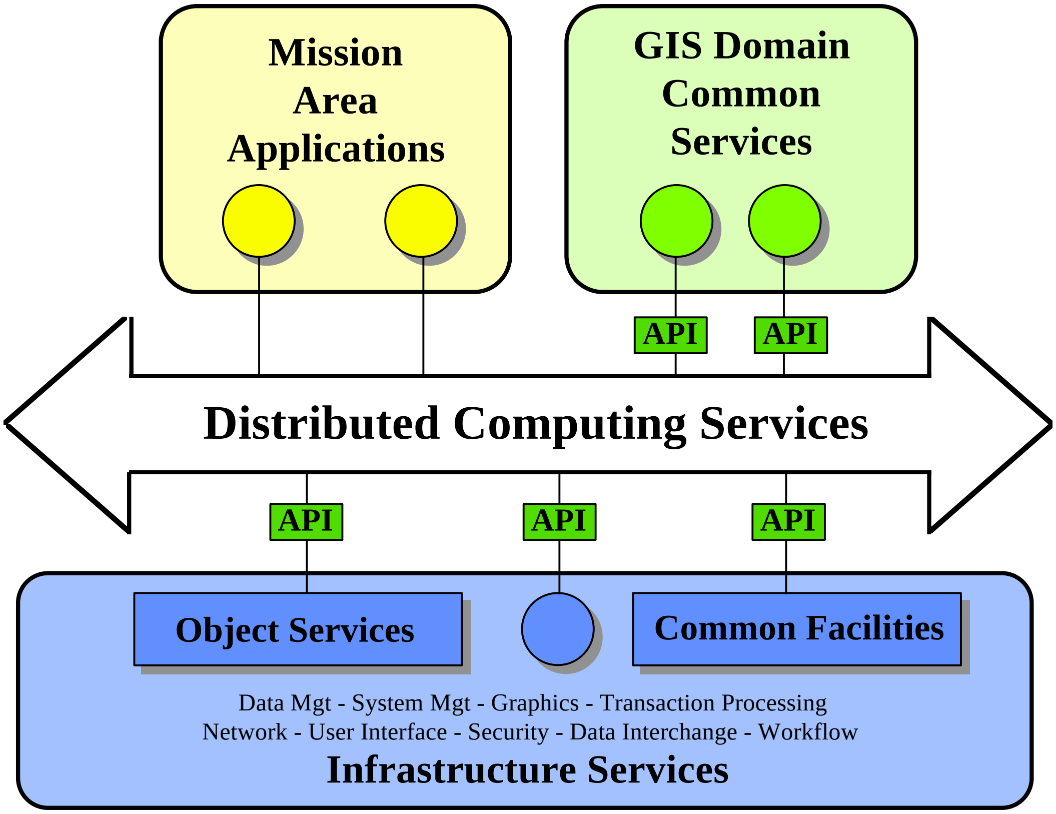
Mission
Area
Applications
GIS Domain
Common
Services
API
API
Distributed Computing Services
API
API
API
Object Services
Common Facilities
Data Mgt - System Mgt - Graphics - Transaction Processing
Network - User Interface - Security - Data Interchange - Workflow
Infrastructure Services

Figure 3-3 USIGS Software Reference Model

The USIGS Software Reference Model shown in Figure 3-3 is adapted from the Object Management Group's Object Management Architecture (OMA). It classifies the components, interfaces, and protocols that comprise an object system. The software model represents the long-term goal of a fully-realized object oriented environment. It is envisioned that as the technical architecture migrates towards purely object oriented designs, the systems architectures will evolve towards compliance with this model. The Infrastructure Services identified in the Technical Reference Model (see Figure 3-1) are anticipated to be replaced by Object Service Interfaces and Common Facilities. The model has six key components:

- **Distributed Computing Services** – Enables software objects to make and receive requests and responses within a distributed environment
- **Object Service Interfaces** – A collection of fundamental services (interfaces and objects) that provide basic functions for using and implementing other software objects
- **Common Facilities** – A collection of higher-level services that are broadly usable by many applications
- **GIS Domain Common Services** – Standard interfaces that promote object-based interoperability within the imagery and mapping community or application domains

- **Mission Area Applications** – Software objects specific to the USIGS, including particular commercial products or end-user systems
- **Application Programmer Interfaces** – The collection of available function calls (or other input/output mechanisms) that enable other systems to obtain services from, exchange data with, or otherwise interact with an application program.

At a more detailed level, Common Facilities and GIS Domain Common Services are defined as those interfaces and uniform sequencing semantics that are shared across applications in such a way as to make object-oriented distributed computing applications much easier to create. Common Facilities and Geospatial Information Services comprise both generic facilities and domain-specific specifications. Examples of the kinds of inter-application services provided by Common Facilities and Geospatial Information Services include object cataloging and browsing, help facilities, object rendering, printing and spooling, and objects which implement generic business rules for the imagery industry.

3.1.2 Communications as the Basis for Interoperability

The Common Imagery Interoperability Facilities architecture is based on a network-centric concept of systems interoperability. This concept reflects the growing importance of widely distributed computer interactions for running the modern business or governmental enterprise. It also reflects the growing trend for software vendors to include support for advanced telecommunications capabilities in their operating system and application products.
Figure 3-4 presents a notional depiction of the relationship between the CIIF 's API-based approach to interoperability standards and the kind of layered architecture that lies at the heart of the CIIF's network-centric approach to distributed computing.

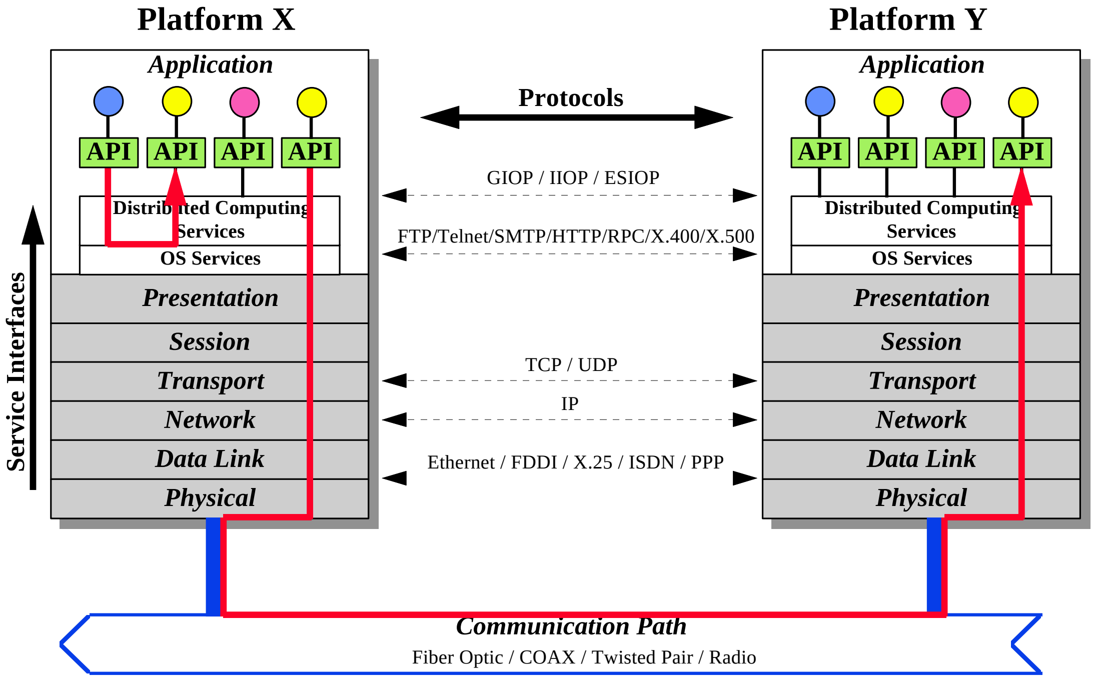

Figure 3-4 CIIF Relationship to the OSI Reference Model

3.1.3 Interface Definition Language (IDL)

The fundamental purpose of this document is to lay the necessary groundwork to support the coordinated development of IDL specifications for the Common Imagery Interoperability Facilities. ISO's IDL was selected for this purpose, because it offers many inherent advantages over older API specification methods.

ISO's IDL is a formal language (similar in appearance to a C++ header file) that is used to define the interfaces between interoperable software objects. Among its chief advantages is that IDL can be directly compiled into any of several common programming languages, using standard language mappings, to automatically set up the mechanisms needed to pass service requests across the network in a distributed software architecture. By adopting IDL to develop the API specifications, the CIIF will capitalize on the many interoperability benefits inherent to the use of IDL including:

- IDL is a widely accepted specification language for modern ICDs. It is actively supported by major standards bodies, such as the International Standards Organization (ISO, Draft International Standard 14750), X/Open, OMG, and OGC. IDL has emerged as an accepted standard for specifying other standards.

- IDL is vendor, platform, and language independent. A single specification suffices for C, C++, Ada, and Smalltalk, which reduces the cost and complexity of documentation, while promoting improved specification rigor.

- IDL is a complete and rigorous interface specification notation. It features strong typing, standard language mappings, exception handling, etc.

- IDL’s standard language bindings enable an IDL specification to be automatically compiled into any of the standard programming languages. The result can be used both to provide an initial framework for software development, and to provide the basis for comparing as-built code to original specifications.

- Can be used with, or independently of, OMG’s Object Management Architecture.

3.1.4 Evolution of Common Facilities and Domain Interfaces

The roles, uses, and definitions of individual Common Facilities and Geospatial Information Services have the potential to evolve over time. Geospatial Information Services that are used extensively in a variety of similar applications may gradually change from being domain-specific to being generic in character. Moreover, services that are offered across multiple application domains are good candidates for incorporation into future versions of the Common Facilities. Discovery of such commonalities will be a favorable indicator of the maturation of these standards.

The boundaries separating Common Facilities and Geospatial Information Services from Mission Area Applications (in the one direction) and from Object Service Interfaces (in the other direction) are therefore not fixed and immutable, but rather, are a reflection of the state-of-the-art in object system technology. As experience in a particular application domain advances, areas of potential new Geospatial Information Services or Common Facilities will be discovered and defined–just as evolving system infrastructures will gradually incorporate pieces of the Common Facilities into their basic Object Service Interface offerings.

Operations provided by the Object Service Interfaces component of the Technical Reference Model are expected to serve as key building blocks for Common Facilities, Geospatial Information Services, and Mission Area Applications. Common Facilities and Geospatial Information Services, in turn, provide higher-level interoperable interfaces that can be specialized for particular Mission Area Applications. The practical application of these various levels of standard interfaces and services makes extensive use of (and in fact, depends upon) the object-oriented concept of inheritance (Figure 3-5). Compiler-based support for these inheritance processes has been built-in to ISO’s Interface Definition Language (IDL), via its various standard language mappings. Inheritance facilitates the standardization of interfaces, promotes interoperability between objects conforming to the base standard, and enables the design of consistent interfaces between otherwise disparate object types.

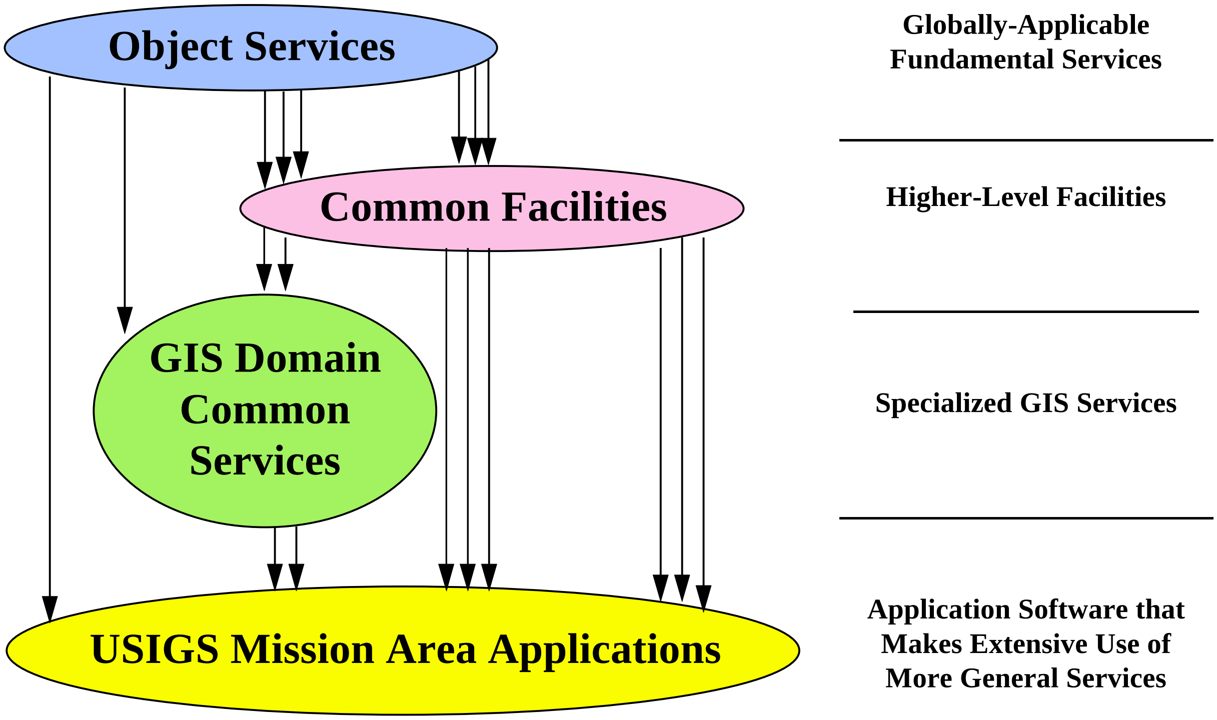
Object Services
Common Facilities
GIS Domain Common Services
USIGS Mission Area Applications
Globally-Applicable Fundamental Services
Higher-Level Facilities
Specialized GIS Services
Application Software that Makes Extensive Use of More General Services

Figure 3-5 Software Component Reuse

In non-object-oriented software architectures, a system's Application Program Interface (API) is typically defined by a fixed, monolithic, interface structure. The IDL-based API that is intrinsic to the CIIF Reference Model, on the other hand, is fully modular–the developer of an application object can pick and choose exactly those interface elements that are needed to fulfill design objectives. In other words, in contrast to the case for the traditional style of API, the object-oriented APIs provided by IDL-specified Geospatial Information Services and Common Facilities are extensible, customizable, and subset-able. This high degree of adaptability will facilitate the controlled evolution of such object-based standards, thereby helping to ensure the long-term viability of the technology.

3.2 Facility Definition Guidelines

This section describes important architectural principles, goals, and conventions that have guided (and should continue to guide) the identification, specification, and development of Common Imagery Interoperability Facilities.

3.2.1   Key Architectural Principles

This section identifies certain fundamental design principles that should be reflected in the overall architecture of a collection of Geospatial Information Services and Facilities:

- Interfaces should be object-oriented and should be defined using ISO IDL. Any part of a facility or domain interface that can be conceived of as being a software object *should* be viewed that way and should be rigorously defined using the ISO IDL.

- Proposed extensions to the CIIF Reference Model or other related standards should be explicitly identified. Facilities should be kept as consistent as possible with the reference model and other applicable standards.

- Operation sequencing should be specified, where applicable. Cooperative behavior among software objects (i.e., the anticipated sequencing of transactions between objects) should be specified as part of the definition of a facility.

- Facility specifications should not include implementation prescriptions; on the contrary, a facility's design specification should seek to minimize implementation constraints, and thereby maximize flexibility. However, provision should be made for application developers to control implementation choices (e.g., by the use of compiler directives or other technical mechanisms that do not affect the facility's functional interface).

- Specifications should be complete, in the sense that all functional capabilities and operations needed to make effective use of a facility should be anticipated and included in its specification. For example, object creation does not just happen, some operation must be called to cause it to occur–life-cycle services should be defined as appropriate to provide these kinds of operations.

3.2.2 Architectural Goals

This section describes a series of architectural goals for both the initial definition and the final specification (in IDL) of the Common Imagery Interoperability Facilities:

- **Maximize the independence and modularity of facilities** – The interoperability-related functionality addressed by the CIIF architecture should be partitioned into a discrete collection of tightly focused, non-overlapping facilities and domain interfaces. Properly accomplished, this partitioning and scope definition process will enable each facility or domain interface to be independently specified and implemented.

- **Minimize duplication of functionality between facilities** – Functionality should be allocated to the most appropriate facility, and each new facility should build on previous facilities, as appropriate. For example, if a "Presentation Facility" has been defined for the purpose of supplying object presentation interfaces, then subsequent facilities (as well as new applications) should use this facility and not reinvent such presentation capabilities for themselves.

- **Avoid using hidden interfaces between facilities** – The interfaces and behaviors of software objects should be openly specified, so that for a particular facility, one implementation can be substituted for another, without affecting the ability of the overall system to continue to function.

- **Maximize consistency among facilities** – Diverse facilities should be able to work together without engendering conflicts or ambiguities. This can be accomplished by using similar (or compatible) interface conventions through-out the CIIF.

- **Promote extensibility of individual facilities** – The design of a facility should naturally accommodate its extension via iterative development processes. The specification for a facility should make clear how such extensions are to be defined (via inheritance, delegation, etc.). Facilities should use the multiple inheritance capability of ISO IDL wherever appropriate, since multiple inheritance eases the task of defining interfaces that address varied purposes.

- **Facilitate extending the list of facilities** – The architecture of a collection of facilities should enable designers to define and standardize new facilities, without having to re-design any system that uses the existing facilities.

- **Provide precise facility descriptions** – A facility's specification should include a precise description of the semantics and side-effects, if any, that govern the use of the facility's operations. For example, a facility's specification should clearly distinguish the interfaces and behaviors provided by that facility from the interfaces and behaviors it expects other facilities to supply.

- **Ensure the integrity, reliability, and safety of facilities** – Safeguards should be provided to guard against the inadvertent corruption of a facility or the objects it manages.

- **Consider performance issues** – Facility interfaces should be designed with performance tradeoffs in mind.

- **Promote scalability** – Facilities should be designed with the expectation that they will have multiple implementations, optimized for an assortment of operating conditions and environments.

- **Promote portability** – Facilities should be designed to accommodate portability of implementations across a wide range of platforms. They should not, for example, require the use of a particular programming language.

3.2.3 Architectural Guidelines and Conventions

One of the keys to producing a robust architecture for the CIIF is to establish architectural guidelines and conventions that promote consistency of style and organization among the Geospatial Information Services. Some of the key issues in this regard include:

- **Inheritance** – The specifications for domain interfaces/facilities should take full advantage of the multiple inheritance feature of ISO's Interface Definition Language. Although each such facility should define a self-contained interface, a particular object implementation might support several distinct interfaces, so multiple inheritance can be used to specify the complete set of operations to be implemented by that object.

- **Exception Handling** – The ISO IDL reference specification defines an exception handling mechanism that returns an exception condition whenever an operation terminates abnormally; that is, for each unsuccessful invocation of an operation, one of a set of pre-defined termination indicators will be returned. Facility specifications should define a complete set of specially-tailored exception conditions, in addition to using the standard exceptions defined in the ISO IDL reference specification. Using IDL's built-in exception handling mechanism will enhance the documentary value of the IDL specifications themselves, and will promote improved error handling by client applications.

- **IDL Naming Conventions** – Facilities and domain interfaces produced by different IDL developers will be much easier to understand and correctly use, if uniform naming conventions are applied to the various components of an interface definition. The following basic conventions are suggested:

  - Tokens representing interface and type names should be capitalized. If the name of an interface or type consists of multiple words, each word should be capitalized.

  - Tokens representing operations, attributes, formal operation parameters, structures, exceptions, and union branches should be typed all in lower-case. If a token consists of multiple words, the words should be separated by underscores (_).

  - Attribute names, definitions, and formats should be taken from the *SPIA* where applicable.

  - Tokens should be titled with meaningful names that clearly represent the object being labeled. The use of reserved words which have meaning to other languages (C, C++, Ada, Smalltalk) should be avoided.

- **Access Controls** – CIIF facilities and interfaces should use a consistent system of access controls to determine who is authorized to use various interface capabilities. A standard mechanism (such as an access control list) is needed, for defining and enforcing permissions to use such capabilities.

- **Other Standards** – To maximize the portability and interoperability of separately developed facilities and domain interfaces, developers should standardize their solutions on POSIX interfaces, where available.

- **Change Management** – As developers gain experience with the CIIF, the nature of the various interfaces and facilities will certainly evolve to, for example, add features, improve performance, or support additional kinds of applications. To minimize disruptions to applications that use the first-generation implementation of the CIIF, therefore, new interfaces should be developed by inheriting from previously adopted interfaces.

3.3 Facility/Interface Design Considerations

This section examines certain technical design considerations that effect the architecture of the CIIF at a more-detailed level. In this regard, a facility (or interface) is characterized by the API it provides, by the types of objects that provide those interfaces, and indirectly, by certain implementation-related characteristics (such as low-level API semantics, or various performance parameters). A facility may involve:

- A single object (e.g., the time-server object)
- Multiple objects, all of which provide the same type of interface (e.g., thread objects)
- Multiple objects, with distinct interfaces that inherit from a common interface
- A combination of these cases.

Each Common Facility or Imagery Interface furnishes its API definitions to a set of users, which are usually Mission Area Applications, but which can also be other Common Facilities or Geospatial Information Services. The latter case can create significant architectural dependencies, as is explained in the following sections.

3.3.1 Interfaces that Comprise a Facility

A facility may have several distinct interfaces (i.e., it may define multiple semantically-related interface types). A taxonomy of these interfaces is presented here, because it is important in characterizing facilities to clearly distinguish what interfaces are involved in providing a facility, how they relate to each another, how one gets access to them, and who is expected to use them. The interfaces to a facility can be characterized by:

- **Audience** – The types of the anticipated consumers (callers) of an interface. An interface may be intended for use by the ultimate user of the facility or it may be intended for use by a system management function within the system. In more complex facilities, objects whose function and implementation lie completely outside of the facility may need to collaborate to fulfill the original facility’s functions. In this case, interfaces may be defined that are used to construct the facility from a series of disparate objects. The audience for such interfaces is neither the user of the facility nor a system manager, but rather the other objects that participate in creating the facility.

- **Bearer** – The object type that presents an interface. An object may be fundamentally characterized by the fact that it has a given interface, or an object may have an interface that is ancillary to its primary purpose (in order to provide certain other capabilities).

The term *audience* characterizes who (or what) uses the different interfaces that comprise a facility. Such interfaces can be categorized as belonging to one of three classes according to their intended audience:

- Interfaces that define the operations invoked by the primary consumers or users of the facility are called Functional Interfaces. These interfaces present the functionality (the useful operations) of the facility. A given facility may have several functional interfaces to provide different aspects of its overall collection of services.

- Interfaces used to communicate with system management services and facilities are called Management Interfaces. These interfaces handle operational control of a service (e.g., setting threshold levels), as well as its installation and deployment (e.g., starting and stopping a service).

- Interfaces that define the operations used to communicate between the core of a facility and related objects that participate in providing the service are called Construction Interfaces. These interfaces are typically defined by the facility, and then inherited and implemented by participants in the facility. In other words, these interfaces are invoked by the facility provider itself. Objects that participate in a facility must support these interfaces. A given facility may have several construction interfaces to connect various parts of its implementation.

The term *bearer* characterizes the objects which present a particular interface. The bearer of an interface can be further categorized according to whether that interface defines the core function of the object (i.e., a specific object bears the interface) or whether that interface defines additional capabilities for an object whose core purpose is something else (i.e., some generic object bears the interface); that is,

- Specific objects can bear an interface. By a specific object it is meant an object whose purpose for existence is to constitute that part of the facility whose interface it carries. The notion is that a limited number of implementations (and potentially a limited number of instances) of these objects exist in a system, usually as “servers.”

- Alternatively, generic objects can bear an interface. In this context, a *generic object* is an object whose primary reason for existence is unrelated to the facility whose interface it carries. The notion is that the facility is provided by having any of several other object types inherit and implement that facility’s interface.

Note that generic and specific bearers of an interface are distinguishable concepts. If a user wants to know if an object is “transactionable,” he issues a query against the type tree. On the other hand, to find out if “naming” is available, the user does not check the type hierarchy, but rather checks the system configuration to see if a “naming server” is running.

3.3.2 Quality-of-Service for a Facility

Quality-of-service objectives are an important consideration during system design and development in general, so not surprisingly, such considerations also have an important role to play in the design and development of interfaces and facilities. Key quality-of-service factors include performance, availability, integrity, safety, and many others–too many, in fact, to be analyzed in any detail in this document. The general principles associated with such considerations are described in this section; the details will have to be addressed during the design and development of each CIIF domain interface and facility.

A particular interface or facility can be implemented in a variety of ways in order to selectively optimize one combination of performance priorities (or other quality-of-service factors) over another. Although the form of the interface should remain unchanged by such trade-offs, the underlying semantics of the interface *can* be affected. It is very important, therefore, that interface specifications be designed in such a way that they can support the quality-of-service priorities that a given implementation is anticipated to be expected to deliver. For example, can a particular interface design deliver high transaction throughput, or does the design constrain the trade-off options in a way that it is likely to be very expensive in many implementations?

Although these kinds of architectural and design issues can be addressed in a variety of ways, the solutions all tend to be rather technical in nature, as is illustrated in the following examples:

- One approach for dealing with potentially demanding performance requirements would be to permit a facility's implementation to be split into a low-latency part and a latency-tolerant part. The former might reside "within" the user of the facility (or at least on the same network node), while the latter part could reside anywhere within the distributed system. The design for a performance-critical interface could also specify the optional use of caching or other well-known performance enhancement techniques. Another way to specify such an architecture might be to explicitly design a facility as a group of objects, some of which can be implemented within the using application. Alternately, such partitioning can be completely hidden within the structure of the facility.

- An approach for dealing with availability issues would be to explicitly consider how objects are to be bound to a facility and whether the facility interfaces are designed in such a way that alternate instances of the facility can take over for a failed instance. Once again, provision for such a capability can be made explicitly visible to the using application (by providing a failure indication, along with an interface that permits continuation of the service elsewhere), or the capability can be hidden behind a portion of the facility that exists within the using application (which therefore cannot fail, unless the using application itself fails).

3.4 Application of Interface Standards

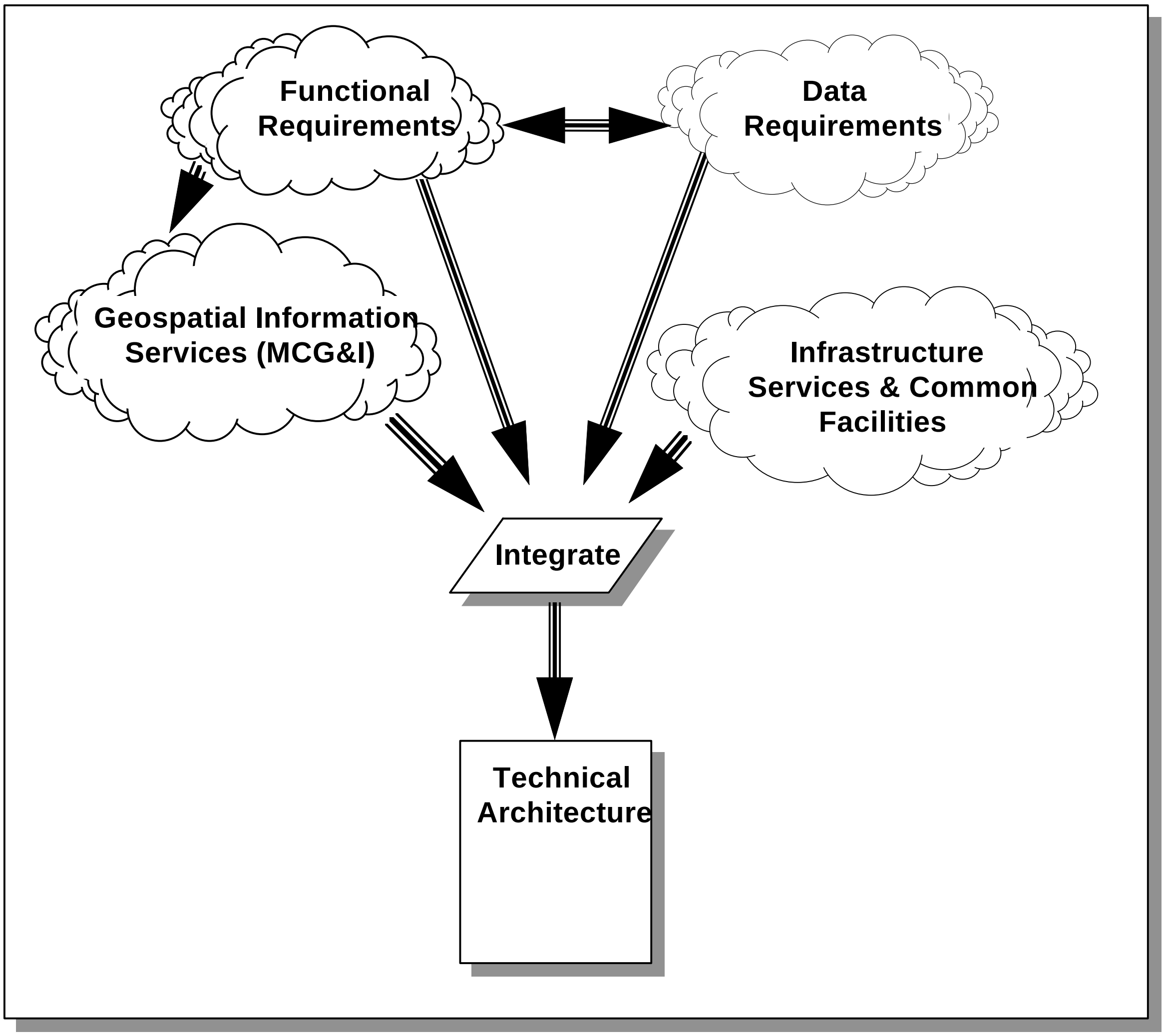
Functional Requirements
Data Requirements
Geospatial Information Services (MCG&I)
Infrastructure Services & Common Facilities
Integrate
Technical Architecture

Figure 3-6 Origins of Interface Standards

A common interface specification definition process derives its detail from several sources. A business process analysis of an enterprise will include the enterprise mission, information flow and concept of operations, and consist of a set of services. This process will generate functional requirements needed to accomplish a particular service. These requirements are extensible and not predicated upon technological implementations. In this instance the service would generate information objects composed of data. These services are then decomposed into those specific for a particular functional service such as MCG&I, and those provided by supporting infrastructure. The infrastructure services would be common across enterprise domains. Additionally data requirements based upon user information needs provide the framework for the information requirements of the functional services, but also provide detail for the interface specification. Once documented, this common interface specification will be described and identified in an enterprise technical architecture, and further refined in profile documents.

Section 4
THE CIIF ARCHITECTURE

4.1 Structural Organization

A technical reference model is a high-level representation of a system’s architecture that enables people to agree on definitions, build a common base of understanding, and identify and resolve crucial issues. A technical reference model is not a specific system design, but rather, establishes a standard vocabulary for the conceptual modules, services, and interfaces that comprise an architecture, in order to provide a context for analyzing portability, scalability, interoperability, and other technological issues. The blocks in the diagram below (Figure 4-1) depict the Digital Elements that comprise the USIGS Technical Architecture, while the “network-bus-style” arrows depict major categories of interfaces that are used to invoke services and move data within the USIGS. This diagram focuses on categorizing the interfaces that are *key* to achieving full interoperability among and within the elements that comprise the USIGS Technical Architecture, and hence, can be viewed as providing a graphical overview of the *CIIF Reference Model*.

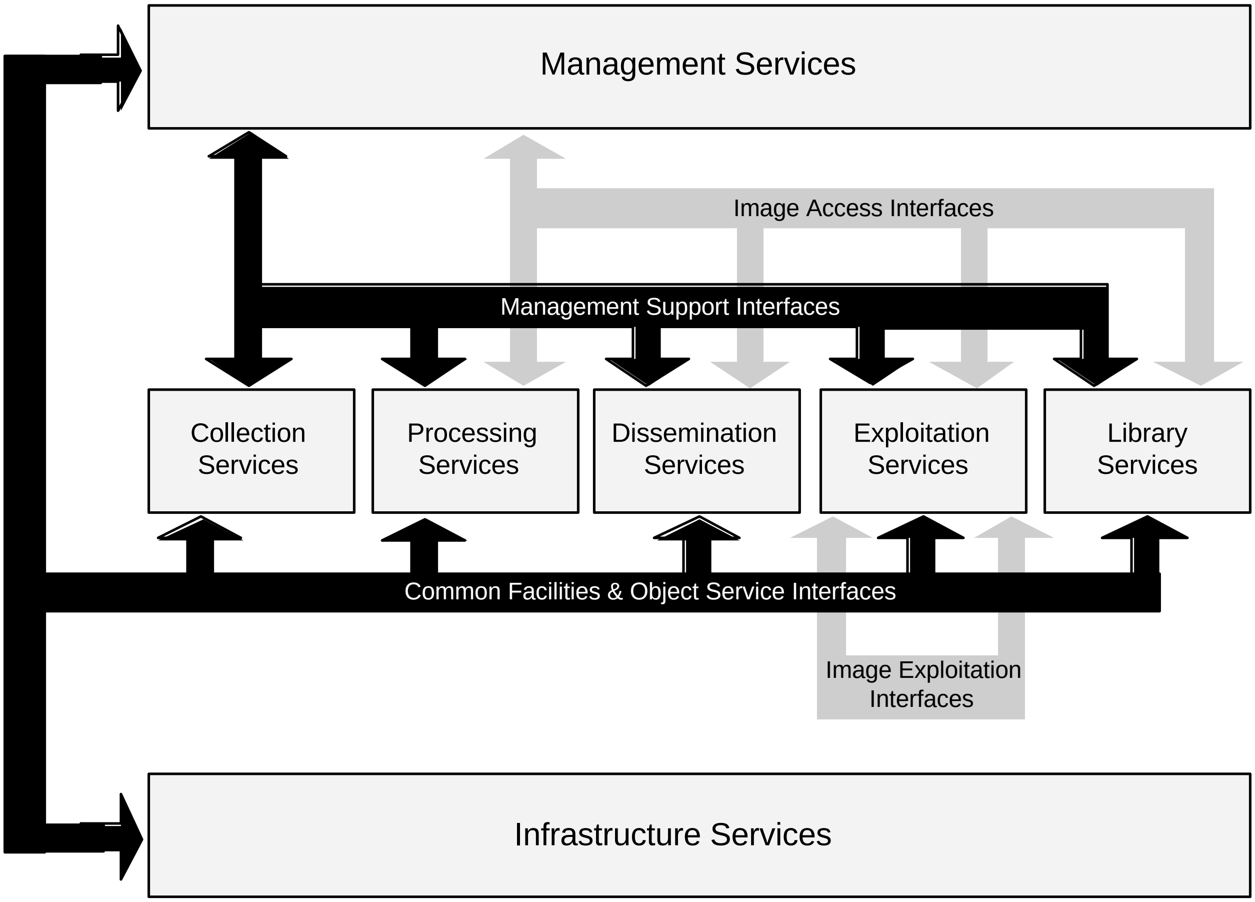

Figure 4-1 Major Categories of USIGS Interoperability Services

As indicated in Figure 4-1, the CIIF Technical Reference Model groups the interoperability interfaces in the USIGS Technical Architecture into four major categories:

- **Image Access Interfaces** – Provides the means to store, catalog, discover, and retrieve imagery and imagery-derived products; define standing profiles of users' imagery-related intelligence interests; and support the automatic dissemination of current imagery and imagery-derived products to appropriate recipients.

- **Image Exploitation Interfaces** – Provides a collection of image manipulation and automated image analysis capabilities, in support of critical imagery exploitation functions.

- **Management Support Interfaces** – Provides improved (more-fully interoperable) capabilities to manage and distribute imagery collection requirements and the associated collection, processing, and exploitation tasking and status reporting; also provides improved capabilities to distribute collection system coverage forecasts, and to manage detailed exploitation tasking.

- **Common Facilities and Object Service Interfaces** – Identifies selected commercial standards for multi-media and compound-document data exchange, for data encoding and compression/decompression support, for imagery display and printing support, and for system security; also identifies potential commercial standards for collaborative processing, generic data base search and retrieval, archival (hierarchical) storage management, and automated negotiation of compatible interfaces (i.e., Service Trading). Object Service Interfaces provide for application consistency and help to increase programmer productivity.

To complete the overview of the structural organization of the technical reference model for the CIIF, Table 4-1 (on the following page) shows how the interface categories described above have been partitioned into various facilities.

Table 4-1 Geospatial Information Services

	Facilities	***Candidate Facilities***
Image Access Interfaces		
	Catalog Access Facility	
	Image Access Facility	
	Imagery Dissemination Facility	
	Profile and Notification Facility	
Image Exploitation Interfaces		
	Image Mensuration Facility	
	Image Manipulation Facility	
	Image Registration Facility	
	Geolocation Facility	
	Image Annotation Facility	
		Automatic Target Recognition (TBR)
		Image Synthesis (TBR)
		Image Understanding (TBR)
Management Support Interfaces		
		Requirements Management Support (TBR)
		Exploitation Management Support (TBR)
		Dissemination Management Support (TBR)
Mapping, Charting and Geodesy Interfaces (TBR)		

4.2 Geospatial Information Services

The fundamental goal of the United States Imagery and Geospatial Information System (USIGS) is to provide an enhanced and better integrated set of services for accessing and managing imagery, imagery-derived intelligence products, and geospatial (mapping) data. To ensure that these services will be well integrated and fully accessible across the DoD’s emerging distributed computing architecture, a standard set of Geospatial Information Service APIs is needed. This section describes the scope and composition of these APIs.

4.2.1 Image Access Interfaces

The ability to interoperably store and retrieve imagery and imagery-derived data, combined with greatly enhanced abilities to search for these data, are fundamental to the architecture of the objective USIGS. Analysis of the *USIS Technical Architecture Requirements*, coupled with an effort to apply the CIIF’s distributed computing architecture principles, has led to the definition of the following facilities:

- **Catalog Access Facility** – Supplies a set of common software interfaces to support both local and global imagery product discovery, product attribute (metadata) retrieval, product browsing, and product cataloging and indexing

- **Image Access Facility** – Defines a set of interfaces for retrieving selected imagery products, from an imagery library, and for updating the contents of an imagery library (by storing, deleting, or modifying imagery products)

- **Imagery Dissemination Facility** – Defines the interfaces required to receive, prepare (i.e., reformat, compress, decompress, etc.), prioritize, and transmit imagery products; also defines standard interfaces to support product distribution management

- **Profile and Notification Facility** – Supplies a set of standard interfaces to support the registration and maintenance of standing interest profiles for imagery consumers; also provides interfaces to support the screening of products against these profiles, and to route products or product availability notifications, as appropriate.

These interfaces are intended to apply to all manner of imagery, imagery products, and imagery-related metadata. The goal is to provide uniform interface mechanisms for handling monochrome still images, color images, stereo pairs, multi-spectral images, full-motion video, co-registered images, maps, and graphics, three-dimensional (solid) visual models, as well as various combinations of these.

The following sections (those that start with the numbers 4.2.1.1) reflect the facilities that have been developed to date and documented in the *Image Access Service Specification* (IAS). The various interfaces comprising each facility are described, and short definitions of the individual operations are provided. Where applicable, USIGS technical architecture requirements that have not yet been implemented by the IAS are discussed in the "Technical Issues" sections.

The interface hierarchy depicted in Figure 4-2 shows the interfaces defined in the IAS. The white boxes signify interfaces, and the associated operations within those interfaces. Arrows indicate the path of inheritance from one interface to another.

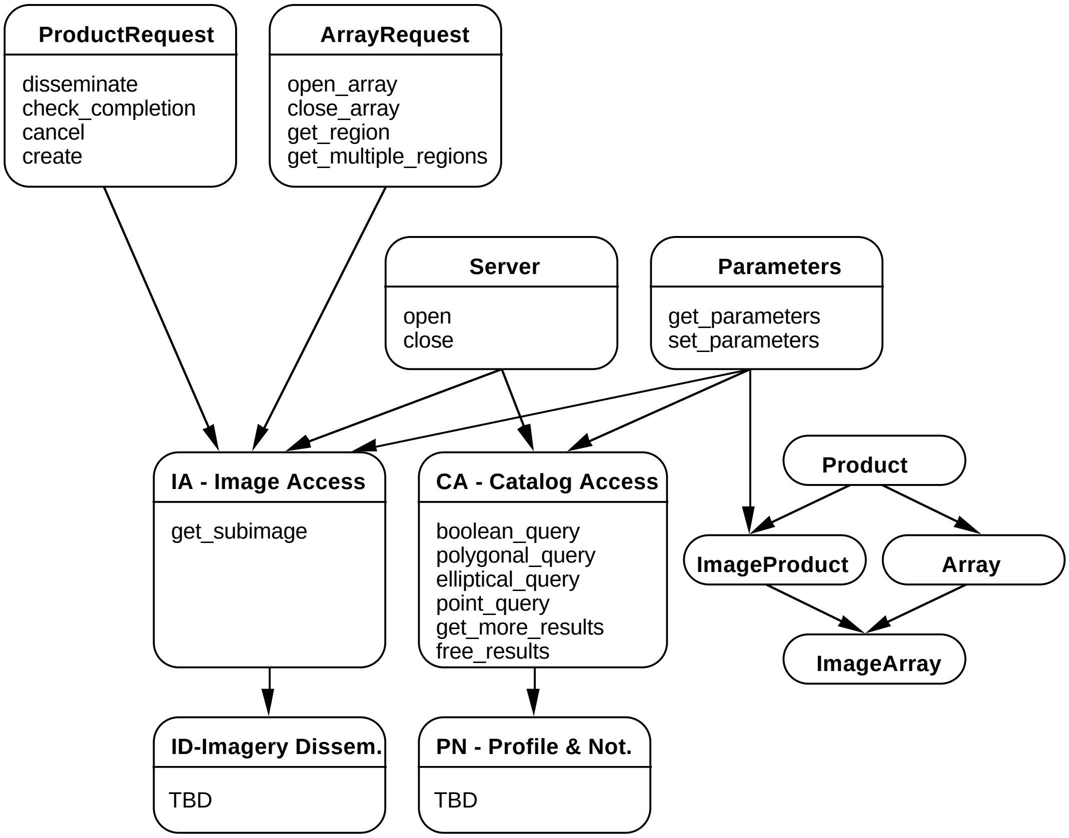

Figure 4-2 IAS Interface/Facility Allocations and Inheritance Hierarchy

4.2.1.1 Catalog Access Facility

The Catalog Access Facility provides a common software interface between an imagery library catalog and client applications for the purposes of product discovery, product attribute (metadata) retrieval, product indexing, directory maintenance, and uniform access to database resources. The Catalog Access Facility enables imagery searches, imagery browsing, and catalog updates.

The Catalog Access Facility inherits the “open” and “close” operations from the Server Interface and the “get_parameters” and “set_parameters” operations from the Parameters Interface.

4.2.1.1.1 IDL Interfaces

4.2.1.1.1.1 Catalog Access Interface

- **Boolean Query Operation** – Enables ordinary catalog search queries by accepting Boolean Query Syntax expressions as input and returning a set of query hits matching the expression.

- **Polygonal Query Operation** – Enables ordinary catalog search queries by supplementing Boolean queries with the specification of a polygonal shape. Image products which overlap any portion of the polygon will satisfy the query if the product attributes also satisfy the Boolean query expression.

- **Elliptical Query Operation** – Enables Boolean catalog search queries combined with the specification of an elliptical shape. The ellipse is defined by its center point, major and minor axes, and the azimuth rotation from North of the major axis and returns image products which provide coverage of any portion of the ellipse while satisfying the Boolean Query Syntax expression.

- **Point Query Operation** – Supplements Boolean queries by specifying a geographic point. Image products returned by this query combine the Boolean Query Syntax expression and coverage of the specified point.

- **Get More Results Operation** – Accesses invocations of Catalog Access Facility operations unable to return entire sets of query hits in the allocated area for query results. A QueryId value is returned and used with the “get_more_results” operation to obtain the remaining query hits.

- **Free Results Operation** – Notifies the catalog server that the client does not intend to retrieve additional results for the indicated QueryId. This enables the catalog server to free any resources allocated to the indicated QueryId, including any remaining results.

4.2.1.1.2 Technical Issues

- The identity of the metadata elements that will be used to catalog imagery products (particularly some of the newer products, such as video) is not fully settled. The logical structure that is used to organize these metadata elements, as well as the data modeling notations used to define the logical structure of their inter-relationships, is not settled either.

- Various enhancements of this facility have been proposed to provide support for cataloging video imagery. For example, cataloging of both video clips and individual frames should be accommodated, as well as queries and metadata that recognize the temporal dimension unique to video imagery.

- The potential applicability of the OMG Object Query Service to fulfill image access discovery requirements has been raised as an issue.

- The ability of the current version of the facility to provide a comprehensive catalog of worldwide imagery product holdings (i.e., a cataloging capability that encompasses all imagery libraries) has been questioned. In particular, a potential need to standardize interfaces for apprising catalog of new product accessions at remote libraries has been identified. Such an interface would need to accommodate free-text (keyword) indexing, as well as standard metadata cataloging of products.

- The current version of the facility lacks any sort of support for maintaining and using topical classification hierarchies of global library holdings. Such a capability would be similar to the “Net Directories” that are currently accessible via browsers on the World Wide Web.

4.2.1.2 Image Access Facility

The Image Access Facility provides a common software interface between a library and its client applications in support of imagery product retrieval. The facility enhances interoperability among high-performance image handling applications by defining standardized methods for storing and retrieving imagery and imagery-related data within a shared library. The Image Access Facility provides a robust and comprehensive set of interfaces for the retrieval of imagery objects (either in whole or in part) with resettable parameters. It may be used directly with an object, or indirectly through a particular client’s interaction with an object.

Digital photographic, video, geographic, and drawing applications produce the tightly-integrated, multi-media style of the Image Access Facility. The Image Access Facility is particularly flexible in terms of accommodating complexity and changes in the definition of the data structures combining images with character-based descriptive data elements in the various kinds of imagery objects produced and used by these applications.

4.2.1.2.1 IDL Interfaces

4.2.1.2.1.1 Product Request Interface

- **Disseminate Operation** – Requests the initiation of the transfer of a whole product. The request may include destinations for the request client and third-party clients, thus this operation supports a form of push-mode transfer through a third-party request. The disseminate operation generates a unique request identifier to be used for tracking the progress of the request using the check completion operation.

- **Check Completion Operation** – Enables clients to check the status of a request. Each request is uniquely identified by a request identifier created as a result of the initiation of the disseminate operation.

- **Cancel Operation** – Enables clients to abort an outstanding request initiated by other operations based on the request identifier.

- **Create Operation** – Stores a new product in the library and generates a new product reference.

4.2.1.2.1.2 Array Request Interface

- **Open Array Operation** – Initiates access to an array object. A call to “open_array” is required for an array product before other operations using the array reference can be invoked.

- **Close Array Operation** – Terminates access to an array object. The “close_array” operation indicates the client has completed access to an array product.

- **Get Region Operation** – Retrieves element (i.e., pixels or tiles) data from an array product. The “get_region” operation identifies the region within the product to be retrieved.

- **Get Multiple Regions Operation** – Retrieves one or more array regions to be placed in memory areas. The “get_multiple_regions” operation identifies several regions within a product to be retrieved.

4.2.1.2.1.3 Server Interface

- **Open Operation** – Enables the client application to open a connection to a server.

- **Close Operation** – Initiates the completion of other invocations to a server interface.

4.2.1.2.1.4 Parameters Interface

- **Get Parameters Operation** – Retrieves attributes directly stored or closely associated with an object, and includes the NameList argument which identifies the parameter names to be retrieved as output arguments.

- **Set Parameters Operation** – Modifies the value of client-specific resettable parameters used to affect subsequent invocations of other operations.

4.2.1.2.1.5 Image Product Interface

The Image Product Interface retrieves and creates whole products as an Image Access Facility specialization interface and allows the client to retrieve some attributes specifically associated with an object without going to the catalog. The Image Product Interface provides robust references to arbitrary information products in a distributed environment through the use of library location information and file path names. Since the product reference contains all of the necessary information for retrieval, distributed processing can be handled transparently. The product references are specialized in the Image Access Facility for use by image libraries and include support for parameters.

The “ImageProduct” Interface inherits both the “get_parameters” and “set_parameters” operations from the Parameters Interface.

4.2.1.2.1.6 Image Array Interface

The Image Array Interface retrieves regions from an image array and is an Image Access Facility generic interface. The Image Array Interface adds attributes needed for the retrieval of an array without going to the catalog. An Array reference corresponds to an image product which contains actual imagery data, suitable for retrieval as regions (i.e., tiles). The array reference refers to a unique image. If several images are contained within an image product, there must be a separate array reference to each enclosed image, although the complete product would have a single ordinary "product" reference. The array references are specialized for use in the Image Access system for use by image libraries and includes support for parameters.

The "ImageArray" Interface inherits both the "get_parameters" and "set_parameters" operations from the Parameters Interface via the Image Product Interface.

4.2.1.2.1.7 Image Access Interface

The Image Access Interface inherits the following operations: "disseminate", "check_completion", "cancel", and "create" from the Product Request Interface; "open_array", "close_array", "get_region", and "get_multiple_regions" from the Array Request Interface; "open" and "close" from the Server Interface of the Image Access Facility; and, "get_parameters" and "set_parameters" from the Parameters Interface. The Image Access Interface has one unique operation, the "get_subimage" operation as described below:

- **Get Subimage Operation** – Generates a subimage and stores it in a specified location. The subimage has the form of a whole image product, although it is not required to be cataloged. The Image Array argument provides the source image product from which the subimage is created. The upper left and lower right arguments define the North West corner of the geographic bounding box, and the Southeast corner of the geographic bounding box respectively. The Location Specification argument defines the destination for the subimage. Each Geography Coordinate structure represents a single geographic location based on floating point degrees of latitude and longitude.

4.2.1.2.2 Technical Issues

- The potential applicability of the OMG Object Persistence Service to fulfill imagery product storage and retrieval requirements has been raised as an issue.

- This facility may need to support additional sub-image extraction mechanisms including:
  - Temporal sub-images of time-sampled imagery, including video clip extraction, frame skipping, and other types based upon frame rates or clip duration
  - Spectral sub-images of color and multi-spectral imagery (MSI).

4.2.1.3 Imagery Dissemination Facility

This facility provides a common software interface between libraries, dissemination services, and client consumers of products. The Image Dissemination Facility enables the formatting, delivery, routing, and prioritization of imagery products, and the tasks associated with product distribution management.

4.2.1.3.1 IDL Interfaces

4.2.1.3.1.1 Imagery Dissemination Interface

The Imagery Dissemination Interface inherits operations via the Image Access Facility as described in the paragraph below. The Imagery Dissemination Interface will enable the formatting, delivery, routing, and prioritization of imagery products, and the tasks associated with product distribution management. Specific Imagery Dissemination Interface operations are to be determined.

This Interface inherits the following operations: "get_subimage", "disseminate", "check_completion", "cancel", and "create" operations from the Product Request Interface via the Image Access Interface; "open_array", "close_array", "get_region", and "get_multiple_regions" operations from the Array Request Interface via the Image Access Interface; "open" and "close" operations from the Server Interface via the Image Access Interface; and, the "get_parameters" and "set_parameters" operations from the Parameters Interface via the Image Access Interface.

4.2.1.3.2 Technical Issues

- The potential applicability of the OMG Object Event Notification Service to fulfill imagery product dissemination requirements has been raised as an issue.

- The potential applicability of the newly identified OMG Control and Management of A/V Streams Service to fulfill video-unique dissemination requirements, such as playback for full-motion video, video clips, and still frames, has been raised as an issue.

4.2.1.4 Profile and Notification Facility

This facility provides a common software interface between dissemination facilities and end-user client applications. This interface enables the Management and Library elements to perform the functions associated with end-user interest profile registration, screening of products versus interest profiles, and initiation of routing products or providing notification of product availability and deletion to clients.

4.2.1.4.1 IDL Interfaces

4.2.1.4.1.1 Profile and Notification Interface

The Profile and Notification Interface inherits operations via the Catalog Access Interface as described in the paragraph below. Specific Profile and Notification Interface operations remain to be determined. The Profile and Notification Interface will provide the functions associated with end-user interest profile registration, screening of products versus interest profiles, and initiation of routing products or providing notification of product availability and deletion to clients.

The Profile and Notification Interface inherits the following operations: “boolean_query”, “polygonal_query”, “elliptical_query”, “point_query”, “get_more_results”, and “free_results” operations from the Catalog Access Interface; “get_parameters” and “set_parameters” operations from the Parameters Interface via the Catalog Access Interface; and, the “open” and “close” operations from the Server Interface via the Catalog Access Interface.

4.2.1.4.2 Technical Issues

- The potential applicability of the OMG Object Event Notification Service to fulfill imagery product receipt notification and profile registration requirements has been raised as an issue.

- The ability of this facility as presently implemented to support global propagation of user profiles has been raised as an issue.

4.2.2 Image Exploitation Interfaces

Imagery exploitation is fundamental to the USIGS. It leads to the generation of intelligence reports and other products which ultimately reach policy makers and other consumers of intelligence. Additional exploitation facilities are likely to be added to this reference model. The following list represents the current proposed categories of facilities for imagery exploitation services:

- **Image Mensuration Facility** – Provides standard interfaces to software tools that are designed to measure the spatial characteristics of objects appearing within images

- **Image Manipulation Facility** – Provides interfaces to standard algorithms for manipulating imagery (resizing, changing color and contrast values, applying various filters, manipulating image resolution, etc.) and for conducting mathematical analyses of image characteristics (computing image histograms, convolutions, etc.)

- **Image Registration Facility** – Provides standard interfaces for automatically aligning, co-registering, or otherwise determining image-to-image spatial correlations on the basis of image content

- **Geolocation Facility** – Defines standard interfaces to software tools that support the derivation of precise geographic coordinates on images and maps

- **Image Annotation Facility** – Provides standard interfaces to software tools that enable symbols, graphics, text, and other media types to be overlaid upon imagery to highlight significant content

- **Automatic Target Recognition** – Provides standard interfaces to software tools that are designed to automatically detect, categorize, count, and determine relationships between objects appearing within images

- **Image Synthesis** – Provides a common software interface for creating or transforming images using computer-based spatial models, perspective transformations, and manipulations of image characteristics to improve visibility, sharpen resolution, and/or reduce the effects of cloud cover or haze

- **Image Understanding** – Enables automated image change detection, registered image differencing, significance-of-difference analysis and display, and area-based and model-based differencing.

The following sections further define *USIS Technical Architecture Requirements* that pertain to this family of facilities.

4.2.2.1 Image Mensuration Facility

This facility provides specifications for interfaces between client applications and services for image mensuration and related functions. This facility will include specifications for interfaces to the following services:

- **Image mensuration** - geometric measurements from monoscopic and stereoscopic imagery, under a variety of acquisition conditions. A Government development activity, RULER, has been defining a standardized set of mensuration algorithms and interfaces over the past several years, and these are expected to provide the basis for this facility. This facility should provide the interfaces needed to perform automatic and interactive measurement of scene or image characteristics, to include the computation of absolute, relative, or transformed coordinates of features in an image, as well as the mensuration of geometric properties, such as lengths, heights, areas, volumes, and orientations of image features.

4.2.2.1.1 Technical Issues

Development of image mensuration interfaces by the Government has been ongoing for some time as part of the RULER program. The Government developed solution may be sufficiently mature that it will not be advisable to expand the mensuration facility to encompass both Government and commercial needs.

Domain-specific mensuration capabilities are also being developed under government programs such as MINT. The associated facilities development is likely to focus on interfaces responsive to the specifics of those government services. It should be recognized that these facilities are likely to continue to evolve in a significant way as the associated technologies develop. Moreover, some of the efforts may involve analysis and quantization which the government will not want revealed beyond the intelligence community.

Additional temporal mensuration functions are needed to exploit video imagery. These functions include the measurement of the speed of objects captured on video, determination of an object's direction of travel, and frame-to-frame mensuration.

4.2.2.2 Image Manipulation Facility

This facility provides specifications for interfaces between client applications and services for digital image processing functions. This facility will include specifications for interfaces to the following services:

- **Fundamental image manipulation** – services to provide image enhancements which increase the analyst’s ability to distinguish between similar appearing areas of a scene, and geometric operations which change the digital image in a controlled way, such that resultant features are displaced from their original position.

- **Advanced domain-specific algorithms** – specialized services not generally applicable within image processing which have applicability for exploitation. Such services are typically government developed (GOTS) rather than commercially available.

- **Domain-specific quantization** – measurement of additional properties contained within imagery. One example is radiometry, the determination of average brightness (radiance) obtained within the area of a pixel, either relatively or absolutely. Radiometric analysis is of value for the exploitation of multi-spectral imagery. Analogous opportunities exist for specialized mensuration, varying according to the characteristics of the sensor and its spectral domain(s).

- **Acquisition conditions analysis** – automatic analysis of unknown or uncertain imaging conditions, including illumination conditions, sensor geometry and state, imaged surface geometry (e.g., stereo analysis), imaged surface composition, and atmospheric effects.

- **Image display manipulation** – image manipulation and display services, including the ability to modify image size, image orientation, pixel resolution, pixel dynamic range, image histogram characteristics, local pixel properties, and image (color) lookup tables.

The image processing domain has been under development for over twenty years. In that time frame, a rich set of techniques and algorithms has been established for image processing primitives. There has been an increase in the power of general purpose computers, including those known generically as workstations, especially over the past ten years. The effect has been to shift the processing of images from predominantly special purpose, shared computers to workstations upon which individual imagery users work.

4.2.2.2.1 Technical Issues

At the end of calendar year 1995, there existed well over half a dozen commercial image processing applications whose value for imagery exploitation has been demonstrated by the government. There are no common standards for image processing among them; each performs its own image processing services and can interchange data with the others on an image file (NITFS) basis. Some vendors are concerned that the adoption of an image manipulation facility for common image processing services will prove onerous. Alternately, such an adoption, if not well thought through, could unduly limit the range of commercial services available for procurement by the government for USIGS implementations.

At the current time, one set of image processing services has become an approved standard; ISO 12087-2; commonly referred to as PIKS (Programmer's Imaging Kernel System). At the end of 1995, there was one implementation only of the services specified in the standard, available from PixelSoft. That implementation set was not an integrated, usable application for imagery exploitation, but rather a tool kit for developers. The PIKS itself is expected to be replaced by an object-oriented implementation, O-O PIKS.

Interfaces to image processing services may need to accommodate some form of scripting that can be used to automatically pre-select certain image processing functions prior to an analyst viewing an image. Users could create a script that captures their personal preferences for viewing an image. For example, an analyst could indicate their desire for images to always be shown with North in a vertical position, zoomed at a specific percentage, cropped to show only a particular area of interest, filtered, etc. All of these features are common to most image processing applications, but how they are invoked may differ. The scripting concept would free an analyst from having to learn all the features of all image processing applications, and thus save valuable time.

4.2.2.3 Image Registration Facility

This facility provides specifications for interfaces between client applications and services for image registration and related functions. This facility will include specifications for interface to the following services:

- **Spatial registration** – registering one image to another, to a map, or other specified data set or coordinate set; correcting for relative translation shifts, rotational differences, scale differences, and perspective view differences.

- **Mosaicking** – merging multiple images of abutting/overlapping coverage to form a composite, single image with greater spatial coverage.

- **Radiometric matching** – matching and blending the radiometric values of corresponding or adjacent pixels from abutting/overlapping coverage to form a smooth visual transition.

- **Multi-image fusion** – combining the information from more than one image into a single imagery product (e.g., combining one image from SPOT 10m. resolution panchromatic coverage with one of LANDSAT 30m. resolution multi-spectral coverage).

- **Rectification** – rectification and orthorectification of images. Rectification removes the effect of obliquity in the image acquisition. Orthorectification additionally removes lateral displacement due to terrain relief.

4.2.2.3.1 Technical Issues

Multiple methodologies exist for image registration. Techniques include the general “rubber sheet” warping process at one extreme, and control point detection and matching at the other end. It is expected that algorithms will range in complexity and the extent of human involvement. A significant amount of flexibility will be required of the interface specifications for spatial registration alone. A similar degree of complexity is expected in developing the specifications for mosaicking, radiometric matching, and multi-image fusion.

Of these services, both mosaicking and multi-image fusion are functions which will depend on spatial registration and radiometric matching services in order to create products from input images.

The creation of mosaics and multi-image products can be regarded as image synthesis; the distinction between these activities and those of the image synthesis facilities may be open to dispute.

Registration and orthorectification may ultimately reside in another facility. These services are geometric corrections to acquired imagery, but somewhat distinct from registration.

If mosaicking of video images is performed, each video frame's original identity and subject metadata must be retained. The temporal qualities inherent to video may also enable temporal registration of imagery.

4.2.2.4 Geolocation Facility (TBR001)

The geolocation facility will define standard interfaces to software tools that support the derivation of precise geographic coordinates of physical features appearing on maps and images. Such coordinates can be defined/derived from various geophysical datums:

- **Rapid Positioning Capability** – interactive point selection to initiate the calculation of the point position according to user-specified coordinate types (i.e., UTM, geocoordinates, and image coordinates).

- **Geolocation** – performs a point location search from user-entered coordinates.

- **Grid Services** – applies standard, custom, or computer generated grids over all imagery types.

- **Coordinate Conversion** – converts geographic coordinates to UTMs and UTMs to geographic coordinates.

4.2.2.5 Image Annotation Facility

These services provide standard interfaces to software tools that enable creation, editing and storage of symbols, graphics, text, and other media types to be overlaid upon imagery to highlight significant content

- **Superimposed Display** – displays annotations superimposed over imagery, and allows for the exhibition of the annotation overlay, the image itself, or both.

- **Symbol Selection** – permits the selection of icons and other symbols from a standardized catalog for placement on an overlay.

- **Image-based Annotation** – retains the annotation elements' position relative to the underlying image regardless of the action or manipulation performed on the image.

- **Image-based Registration** – registers annotations to user-specified points on the display of the image or map graphic.

4.2.2.6 Automatic Target Recognition (TBR002)

This category contains facilities which provide specifications for interfaces between client applications and services related to automatic target recognition. This facility will address the need to interface with a variety of algorithms and to provide parametric guidance for the control, training, feedback, and other aspects of the ATR process:

- **Imagery content recognition** – automatic and interactive detection and counting of objects and relations required for an exploitation task (e.g., perform automatic target recognition).

- **Imagery content trend analysis** – automatic and interactive trend analysis for objects and relations of interest in an exploitation task (to include spatial inference from evidence of as-yet unseen, occluded, or otherwise obscured objects, as well as model analysis using time-series and machine learning techniques).

- **Geometric analysis** – automatic and interactive analysis of sensor line-of-sight, terrain and cultural feature classifications (including standard map features and point target types), vehicle- and unit-level location probabilities, mobility analysis, etc.

4.2.2.6.1 Technical Issues

Automatic target recognition is a technical area largely defined by government development at this point. For this reason, commercial involvement in the development of the facility may be largely limited to those developers with significant involvement in government development activities. Moreover, the facilities developed for ATR may be limited to government use, since more global acceptance of ATR is not foreseen in the near term.

Prototype video imagery exploitation tools are already under development. These tools support the temporal dimension unique to video imagery. Video ATR applications provide the ability to perform object motion characterization (if it moves like this, it's a tank), object behavior characterization (this car tends to operate in this manner), and object relationship characterization (this tank is driving over that car).

4.2.2.7 Image Synthesis (TBR003)

This category contains facilities which provide specifications for interfaces between client applications and services for image synthesis and related functions such as modeling. Image synthesis is the creation of imagery products which are essentially synthetic images. This facility will include specifications for interface to the following services:

- **Object modeling** – create CAD (e.g., PHIGS or PHIGS Plus), or other models of elements and objects within an imaged scene.

- **Synthetic image generation** – create a new image from an existing one to simulate changes in acquisition conditions such as illumination, atmospheric effects, sensor geometry.

- **Image perspective transformation** – create an image for a viewer as though taken from a location/perspective other than that of the original image. Generally, this involves creation of a three dimensional scene model of the scene using multiple data sets, which may include stereoscopic imagery, for elevation information.

- **3D fly-by generation** – create three-dimensional models with a perspective center changing with observation time, as though taken from an aircraft (for example) flying over the scene. This builds upon image perspective transformation capability.

4.2.2.7.1 Technical Issues

As mentioned above, the extent to which some services have interface specifications within this area, versus within Image Understanding, has yet to be resolved. The particular parameterization for facilities within this category may not have been sufficiently addressed to provide a mature, flexible interface at this time. Synthetic image generation is itself a category so rich in possibilities that an early solution interface may be insufficient for later interpretations of the service area.

Frequently, multiple services will be invoked to create synthetic image products. For instance, a request for an image perspective transformation may also include the specification of illumination conditions. The facilities' interactions and dependencies must be addressed during their design.

Capabilities for video frame averaging, used to sharpen or smooth video images, are required.

4.2.2.8 Image Understanding (TBR004)

This category contains facilities which provide specifications for interfaces between client applications and services to image understanding and similar information extraction and analysis techniques. This facility will apply knowledge-based inference techniques to extract intelligence from imagery beyond factual scene content: e.g., applying doctrinal knowledge and terrain analysis along with scene content to infer likely vehicle placement beyond the spatial scene limits. This facility will include specifications for interface to the following services:

- **Change detection** – comparing multiple images taken at different times and highlighting areas where significant change has occurred: e.g., the absence of an aircraft where one previously was parked.

- **Pattern recognition** – performs pattern recognition on an image. Pattern recognition is a function which detects the existence of a learned pattern, such as edges joined in right angles.

- **Object recognition** – provides the capability to identify and classify objects in an image. Object recognition is based on the automated or computer assisted detection of recognized patterns, from which detection of a known class of object–based on prior classification–can be inferred.

- **Feature extraction** – extract features from an image based on object recognition. Feature extraction implies the detection and identification of an object but further includes the symbolization of the feature. For instance, detecting, extracting, and re-symbolizing a feature such as a road into a spatial data base or map.

- **Terrain analysis** – techniques for the display, extraction, and analysis of terrain data. Examples of terrain data of interest are: elevation data, soil types, vegetation classes, and drainage patterns. An example of a terrain analysis service is the use of digital elevation data with imagery to generate obscuration profiles, showing visual or signal obscuration between selected points.

- **Content negation analysis** – automatic and interactive negation (determination of origin) of changes detected in objects and relations (e.g., performs site analysis using knowledge-based analytical methods.

4.2.2.8.1 Technical Issues

Some functionality, such as change detection, is well understood and amenable to development of facilities today. Other functionality is currently not mature operationally within the imagery exploitation community, and so will be more problematic to define in the near term.

The approach taken will be to define the well-understood facilities with current applicability, while providing place holders for others, such as image understanding. If and when there is both a need and a viable solution for such services, the facilities for them will be developed.

Prototype video imagery exploitation tools are already under development. These tools support the temporal dimension unique to video imagery. Video imagery understanding applications provide the ability to perform object tracking and motion extraction.

4.2.3 Management Support Interfaces

Providing enhanced management capabilities to improve the ability to track the collection, processing, and exploitation management workflow is essential to the USIGS. Although facilities to standardize the associated interfaces have not yet been identified or defined, Management Support Interfaces are expected to provide support for the following transactional capabilities:

- **Requirements Management Support (TBR005)** – Enable intelligence consumers to submit imagery nominations and obtain feedback on the status of nominations; also defines mechanisms for the exchange of planning, feedback, status, and operational statistics/status reporting information, and for the re-prioritization and re-planning of individual collection, exploitation, and dissemination requirements.

- **Exploitation Management Support (TBR006)** – Provide mechanisms to exchange exploitation task packages, task assignments, exploitation assignment status data, and exploitation resource availability information.

- **Dissemination Management Support (TBR007)** – Provide standard interfaces for balancing and optimizing the imagery product delivery workloads associated with dissemination processing; also provide enhanced capabilities to inquire about, and track, the status of individual product deliveries.

4.2.4 Mapping, Charting & G eodesy Interfaces (TBR008)

The USIGS geospatial information services API standards will need to include a variety of capabilities for accessing and manipulating maps, charts, and various kinds of geospatial data. In addition to services for finding, retrieving, tailoring, and displaying maps and charts, support for seemless (tile-based) access to geospatial feature data, as well as support for merging such data with various kinds of imagery, will be needed. Provisions will be made for plotting order-of-battle and other general military intelligence data on maps or geospatially-correlated imagery. The details remain to be determined.

4.3 Common Facilities

The following Common Facilities, which are likely to be developed by other organizations such as OMG, fulfill key requirements of the USIGS Technical Architecture. As the IDL specifications for each of these facilities are completed and published, they will be thoroughly evaluated, and those that are found to meet the requirements for the USIGS will be adopted as a component of the CIIF.

- **Automation and Scripting Facility –** Defines conventions and interfaces that allow access to the key functionality of an object from another object. The design goal of this facility is to support user visible objects which are larger grained than the typical ORB object. The typical object acted upon by the Automation and Scripting Facility would be a document, a paragraph, a spreadsheet cell, and so forth. The emphasis of the facility is for objects to expose enough of their capabilities so they may be driven by scripts and macros.

- **Common Management Facility –** Provides a set of utility interfaces for system administration functions. These abstract basic functions such as control, monitoring, security management, configuration, and policies that are needed to perform systems management operations, such as adding new users, setting permissions, installing software, and so forth.

- **Compound Presentation and Interchange Facility –** Enables the creation of cooperative component software that supports compound documents, that can be customized, that can be used collaboratively, and that is available across multiple platforms. Also provides for the storage and interchange of data objects, and roughly maps to the persistent storage subsystem of a compound document architecture.

- **Data Interchange Facility** – Allows for the exchange of information across networks of heterogeneous computer systems by providing a common information model and a common way of encoding information within that model. Encoding must support not only character data, but other sorts of data as well, including imagery, graphics, multimedia documents, and electronic mail. Enables objects to interoperate through exchange of data, and can be used for many forms and kinds of data transfer, such as: bulk data transfer; interchange of formatted data such as TIFF, GIF, EPS, NITF, etc.; structured data transfer such as ISO IDL specified data types; interchange of domain-specific object representations; and the data interchange between objects and encapsulated software (legacy applications).

- **Imagery Compression Facility (TBR009)** – Provides standard interfaces to generalized services for imagery (including video imagery) compression and decompression, and for conversion between internal representations and standardized representations of such data.

- **Information Storage and Retrieval Facility** - Comprises the higher level storage and retrieval specifications for distributed applications. These specifications will be applicable to a wide range of information services, including data base access and information highways.

- **Internationalization and Time Operations Facility** – Enables developers to use an information system or application in their own language using their own cultural conventions. In addition, this technology will allow the developer to use a culture's numeric and currency conventions, and keep track of time zones.

- **Meta-Object Facility –** Defines the interfaces and sequencing semantics needed to create, store and manipulate object schemas that define the structure, meaning, and behavior of other objects within the OMG Object Management Architecture. These objects may be application objects, common business objects, objects representing analysis and design models of applications, or objects providing the functionality of Common Facilities and Object Service Interfaces. The Meta-Object Facility can be used in an information system (such as a repository) that enables an enterprise to specify and manage a wide variety of information assets with a common, integrated set of services. The use of a common meta-object facility for specifying the schemas of the information assets will play a key role in helping to achieve data and process integration by enabling tools and processes to share information and coordinate activities.

- **Mobile Agents Facility –** Supports the need to create massively distributed information systems over Wide Area Networks. Agent technology efforts range from building these massively distributed systems to mobile information systems, intelligent workflow systems, and agile corporation information structures.

- **Printing Facility –** One component of a coordinated set of facilities and standards needed to satisfy the printing requirements of the modern distributed office. Together, the capabilities provided must enable users to create and produce high-quality documents in a consistent and unambiguous manner within a distributed object environment. The Printing Facility should be able to meet a range of printing requirements from simple one document, one copy printing, all the way up to high volume production printing, which might involve several documents, several copies, several printers.

- **Rendering Management Facility** – Provides facilities to present information for output on devices such as screens, printers, plotters and sound and speech output devices. It also handles user input from a variety of hardware devices such as a mouse, keyboard, scanner, speech recognition device, digital camera, and security devices. Rendering management includes support for window management, class libraries for user interface objects, user interface dialog objects, and abstractions of the many different input/output devices.

- **Security Administration Facility** – Provides standard interfaces, as well as the necessary control mechanisms, to facilitate required security protections, including provisions for:

  – User registration, password maintenance, permissions maintenance
  – Access control, authentication, and audit trail maintenance
  – Resource registration
  – Security classification downgrading
  – Encryption key management
  – Discretionary and mandatory access controls.

- **Workflow Facility –** Provides management and coordination of objects that are part of a work process for example, purchase orders. The facility will provide support for production-based workflow, which is structured, pre-defined processes that are governed by policies and procedures, as well as ad-hoc, or coordination-based workflows, which are evolving workflows defined by one or more people to support the coordination of knowledge workers.

4.4 Infrastructure Services

A number of important "enabling" capabilities and services are needed to ensure that the practical implementation of the distributed computing architecture will be economically and reliably achieved. In both the Defense Information Infrastructure Common Operating Environment (DII COE) and the CIIF Reference Model, these enabling capabilities have been allocated to the Infrastructure Services component of the architectural model. Although a number of Infrastructure Services are identified in the DII COE, the Object Services are of particular import in connection with the CIIF architecture. These services are expected to become increasingly important as industry gains experience with object-based distributed computing approaches (such as the OMG's). The following section describes these Object Services in greater detail.

4.4.1 Object Service Interfaces

The following Object Service Interfaces, which are likely to be developed by other organizations such as OMG, fulfill key requirements of the USIGS Technical Architecture. As the specifications for each of these services are completed and published, they will be thoroughly evaluated, and those that are found to meet the requirements for the USIGS will be adopted as a component of the CIIF.

- **Collections Service –** Provides a uniform way to generically create and manipulate the most common collections. Collections are groups of objects which, as a group, support some operations and exhibit specific behaviors related to the collection, such as stacks, queues, and lists.

- **Concurrency Control Service –** Enables multiple clients to coordinate their access to shared resources. Coordinating access to resources means that when multiple, concurrent clients access a single resource, any conflicting actions by the clients are reconciled so that the resource remains in a consistent state. The Concurrency Control Service consists of multiple interfaces that support both transactional and non-transactional modes of operation.

- **Event Service –** Provides basic capabilities that can be configured together in a very flexible and powerful manner. Asynchronous events (decoupled event suppliers and consumers), event "fan-in," notification "fan-out," and - through appropriate event channel implementations - reliable event delivery are supported. Both push and pull event models are supported i.e., consumers can either request events or be notified of events, whichever is needed to satisfy application requirements.

- **Externalization Service –** Defines protocols and conventions for the externalization and internalization of objects. Externalizing an object is to record the object state in a stream of data (in memory, on a disk file, across a network, etc.) and then internalize it into a new object in the same or different process. The externalized object can exist for arbitrary amounts of time, be transported by means outside the ORB, and be internalized in a different, disconnected ORB.

- **Interface Type Versioning Service –** Provides for the management of the evolution of interfaces and their IDL descriptions. The service will provide the ability for a system to recognize that a new interface has evolved from an old one, allow multiple versions of the "same" interface to exist at the same time, allow for a choice of automatic or manually specified conversion to use of a new interface, and support the concept of a default version of an interface.

- **Licensing Service –** Provides a mechanism for producers to control the use of their intellectual property in a manner determined by their business and customer needs. This service offers fundamental usage control.

- **Life Cycle Service –** Defines services and conventions for creating, deleting, copying, and moving objects. Because Distributed Computing Environments support distributed objects, life cycle services define services and conventions that allow clients to perform life cycle operations on objects in different locations.

- **Messaging Service –** Provides interfaces that allow clients to make requests on an object without blocking the client execution thread. Some requests are not expected to be complete during the lifetime of the client execution environment, so mechanisms will be established to receive the response and process it appropriately. The service allows object servers to control the order in which incoming requests are processed.

- **Naming Service –** Provides the ability to bind a name to an object relative to a naming context. A naming context is an object that contains a set of name bindings in which each name is unique. To resolve a name is to determine the objects associated with the name are given context. Through the use of a "names library," name manipulation is simplified and names can be made representation independent thus allowing their representation to evolve without requiring client changes.

- **Persistent Object Service –** Provides common interfaces to the mechanisms used for retaining and managing the persistent state of objects. The Persistent Object Service will be used in conjunction with other Object Service Interfaces, for example, naming, relationships, transactions, life cycle, etc. The Persistent Object Service has the primary responsibility for storing the persistent state of objects, with other services providing other capabilities.

- **Properties Service –** Provides the ability to dynamically associate named values with objects outside the static IDL type system. The interfaces provided by this service are used for defining, deleting, modifying, enumerating, and checking for the existence of properties. By using the interfaces defined by the Property Service, useful information can be associated with an object's state, for example, a title or a date.

- **Query Service –** Provides query operations on collections of objects. The queries are predicate-based and may return collections of objects. They may be specified using object derivatives of SQL and/or other styles of object query languages including direct manipulation query languages. Query operations include selection, insertion, updating, and deletion on collections of objects or data.

- **Relationship Service –** Allows entities and relationships to be explicitly represented. Entities are represented as objects. The service defines two new kinds of objects: relationships and roles. A role represents an object in a relationship. The Relationship interface can be extended to add relationship-specific attributes and operations. Similarly, the Role interface can be extended to add role-specific attributes and operations.

- **Security Service –** Protects an information system from unauthorized attempts to access information or interfere with its operation. For example, security services may include (but are not limited to) the following:

  - Confidentiality: information is disclosed only to users authorized to access it.
  - Integrity: information is modified only by users who have the right to do so, and only in authorized ways. It is transmitted only between intended users and in intended ways.
  - Accountability: users are accountable for their security relevant actions. A particular case of this is non-repudiation where responsibility for an action cannot be denied.
  - Availability: Use of the system cannot be maliciously denied to authorized users.

- **Startup Service –** Enables requests to automatically start up when an Object Request Broker is invoked.

- **Time Service –** Maintains current time, ascertains order in which events occurred, and computes the interval between two events.

- **Trader Service –** Provides a matchmaking service for objects – registers availability of the service, provides parameters, distinguishing attributes, and names of operations to which it will respond. It also allows objects in different domains to negotiate and share services without losing control of their own policies and services.

- **Transaction Service –** Supports multiple models (flat and nested) of transactional behavior in a distributed heterogeneous environment. The Transaction Service brings the transaction paradigm, essential to developing reliable distributed applications, and the object paradigm, key to the productivity and quality in application development, together to address the business problems of commercial transaction processing.

4.5 Security Services

To support timely delivery of imagery and imagery products to all echelons of the intelligence community, the Common Imagery Interoperability Facilities will need to address crucial security issues connected with globally distributed data access. Security is a complex problem in a distributed architecture such as that envisioned for the USIGS. These problems are compounded when the architecture crosses multiple security policy domains.

There are no existing security-related government regulations or guidance for object-oriented systems such as CORBA. Traditional security guidance can be applied to the DCE paradigm, however, no government standards exist for DCE security services. Thus, the security services described in Table 3-1 represent an initial attempt at interpreting existing security regulations and guidance to propose standard security services for these new paradigms.

A significant operational requirement for USIGS has driven the need for additional security services that are not adequately addressed in existing security regulations. The need to distribute imagery products to possibly rapidly changing communities of interest has driven the concept of a limited form of label-based interdomain confidentiality service for system-high domains where, normally, security labeling would not be required. Labels are required only on imagery or imagery products that must be shared or disseminated to other domains. This section briefly describes these and other security considerations that pertain to the interoperable architecture of the USIGS.

For a more detailed discussion of these concepts and services, along with other important guidance such as assurance, see *CIIF Security Services Analysis,* NEL, 30 September 1996.

4.5.1 Security Policy Domains

A key concept in the formulation of the USIGS security services is the idea of a security policy domain. A security policy domain provides for threat attenuation to an acceptable level of risk, as determined by the owner of the information in the domain. Each domain consists of information and a system infrastructure that processes and stores the information. Each security policy domain is separately certified and accredited as an operational capability having acceptable levels of performance and security risks.

The traditional concept of security policy domains is shown in Figure 4-3. A system can be accredited to operate in system-high, compartmented, or multilevel security modes. For the most part, compartmented mode requires the same features as multilevel mode. Interoperation among several domains can be approved. However, all interactions have to be pre-determined. Interconnection of these domains can only occur after each domain is accredited and a Memorandum of Understanding (MOU) is approved by the appropriate accrediting authorities.

A system-high domain must enforce need-to-know, commonly enforced by identity-based access control. Typically, most systems implement Discretionary Access Control (DAC). System-high accredited systems are common today.

Multilevel systems, on the other hand, generally employ hierarchical label-based access control (e.g., Mandatory Access Control [MAC]). Generally, all data instances are labeled, and that label is used in mediating access control requests. These systems tend be impractical and expensive, because of the high-level of developmental assurance and performance overhead required. Thus, multilevel systems should not be considered for all USIGS domains. However, this does not mean that multilevel systems should never be considered for some parts of USIGS.

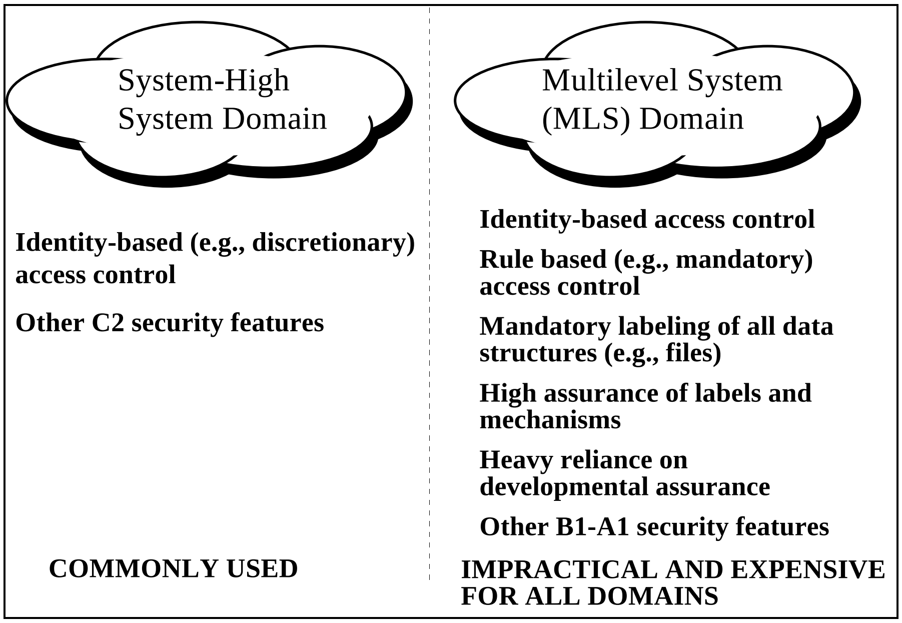
System-High System Domain
Identity-based (e.g., discretionary) access control
Other C2 security features
COMMONLY USED
Multilevel System (MLS) Domain
Identity-based access control
Rule based (e.g., mandatory) access control
Mandatory labeling of all data structures (e.g., files)
High assurance of labels and mechanisms
Heavy reliance on developmental assurance
Other B1-A1 security features
IMPRACTICAL AND EXPENSIVE FOR ALL DOMAINS

Figure 4-3 Traditional Concept of Security Policy Domains

The USIGS concept of Security policy domains, shown in Figure 4-4 , is based on multiple, mutually suspicious, dynamically communicating domains. Initially, these would probably be accredited as system-high domains. Each domain would need to enforce need-to-know internally, using some identity-based access control. Only large-grained objects, typically identified as imagery products required to be imported/exported, would be labeled. The concept requires integrity of the data label so it can be trusted, as well as the existence of an interdomain confidentiality service that enforces an import/export security policy.

It is essential that interoperation among domains be allowed to occur in a dynamic fashion. It will not be possible to predetermine all future possible imagery consumers or their needs. Each domain must be able to configure dynamic communities of interest that are authorized to request imagery products based on need-to-know and the import/export policy. The domains must be mutually suspicious of each other, requiring mutual authentication, intrusion detection, and an exchange of USIGS standard security attributes and other policy information to negotiate interoperation between any domains.

The concept of mutually suspicious negotiation should not be complex to implement if all domains use the USIGS standard security services. Thus, the USIGS concept should be more feasible to implement in the near term than the traditional multilevel system. Figure 4-4 is high-level, showing the minimal services that are needed. This brings up two basic issues. The first issue involves answering the questions of what security services are needed to enforce security policies internal to a domain.

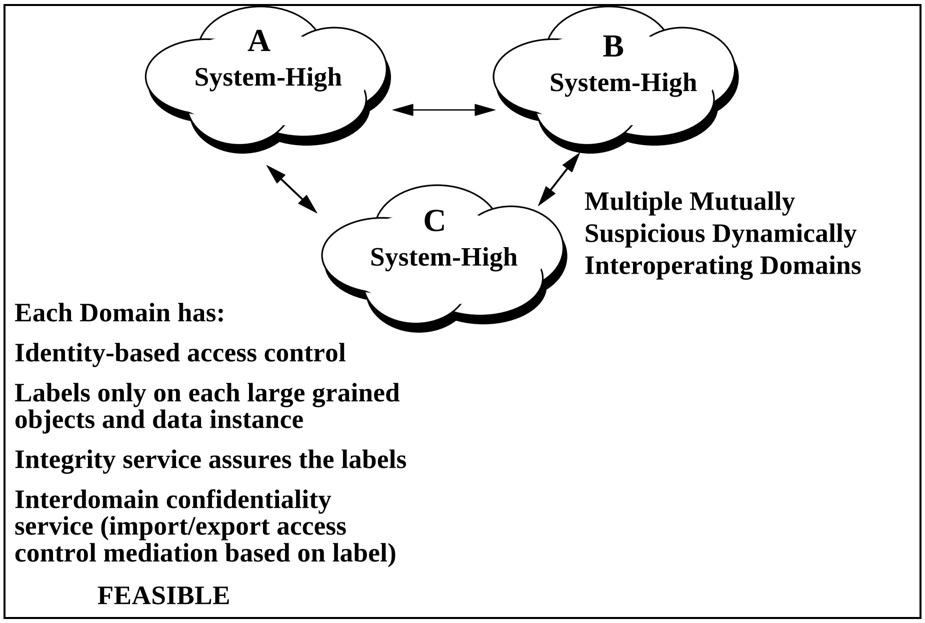
A
System-High
B
System-High
C
System-High
Multiple Mutually
Suspicious Dynamically
Interoperating Domains
Each Domain has:
Identity-based access control
Labels only on each large grained objects and data instance
Integrity service assures the labels
Interdomain confidentiality service (import/export access control mediation based on label)
FEASIBLE

Figure 4-4 USIGS Concept of Security Policy Domains

The second issue addresses the question of what services are required in order to securely interoperate across security policy domains. USIGS has a mandate in operational environments to share certain large-grained softcopy imagery products across many possible domains. Only specifically identified products would be shared based on an import/export policy. Therefore, the import/export policy determines to what extent one domain is able to communicate and interoperate with other domains. Policies may evolve over time; for example, products may be downgraded or declassified. Therefore, the enforcement mechanisms need to be flexible.

4.5.2 USIGS Standard Security Services

The analysis of existing national directives, the USIGS architecture and reference model, and national and international security standards resulted in the derivation of a set of security services that may be needed by USIGS. Figure 4-5 summarizes these services. For the purpose of brevity, not every service is addressed. However, interdomain confidentiality issues and the resulting services are significant enough to warrant discussion.

- **Accountability**
  - Identification & Authentication
  - Auditing
    - Audit Event Server
    - Audit Repository & Processing Facility
    - Intrusion Detection Facility
- **Intradomain Confidentiality**
  - Imagery Labeling
  - Informational Labeling
  - Relabeling Support
  - Access Controls
    - Rule-Based
    - Identity-Based
    - Delegation with Use Specific Auditing
- **Other Services**
  - End-to-End Transaction Compromise Protection
  - Communications Data Compromise Protection
  - Selective Routing
- **Interdomain Confidentiality**
  - Interdomain Security Gateway
    - SIIOP Firewall Proxy
    - Interdomain Interceptor
    - Domain Validation
    - Message Validation
    - Imagery (Label) Validation
    - Non-Imagery Data Validation
    - Interface (Service Request) Validation
    - User Credential Validation
  - Domain Image Server Proxies
- **Integrity**
  - Shield
  - Unshield
  - Validate
- **Non-Repudiation**
  - Evidence Generation and Verification
  - Evidence Storage and Retrieval
  - Delivery Authority
- **Operational Assurance**
  - Security of Management Functions
  - System & Network Management
  - Security Management

Figure 4-5 Summary of USIGS Standard Security Services

4.5.3 Interdomain Confidentiality

A major new service is interdomain confidentiality, as illustrated in Figure 4-6. This service must enforce a domain import/export policy that specifies what services and data instances can be accessed by dynamic request from an external domain. The service can be thought of as an interdomain import/export access control.

At the heart of the service is an interdomain message interceptor that intercepts distributed system message traffic flowing between one domain and all other domains. The request is mediated before being relayed to the specific internal server (for example, a digital image library server).

A label-based import/export policy can make this mediation decision simple. However, considering performance issues, a labeling service is needed to provide a label to only large-grained objects, such as a softcopy image product. An integrity service is needed to prevent unauthorized modification of the label from within the domain. This concept depends heavily on operational assurance for management and mitigation of security risks. Standard interfaces for the listed facilities are required so that vendors can build products that help manage the security risks. These products should be based on standard capabilities used through the standard interfaces.

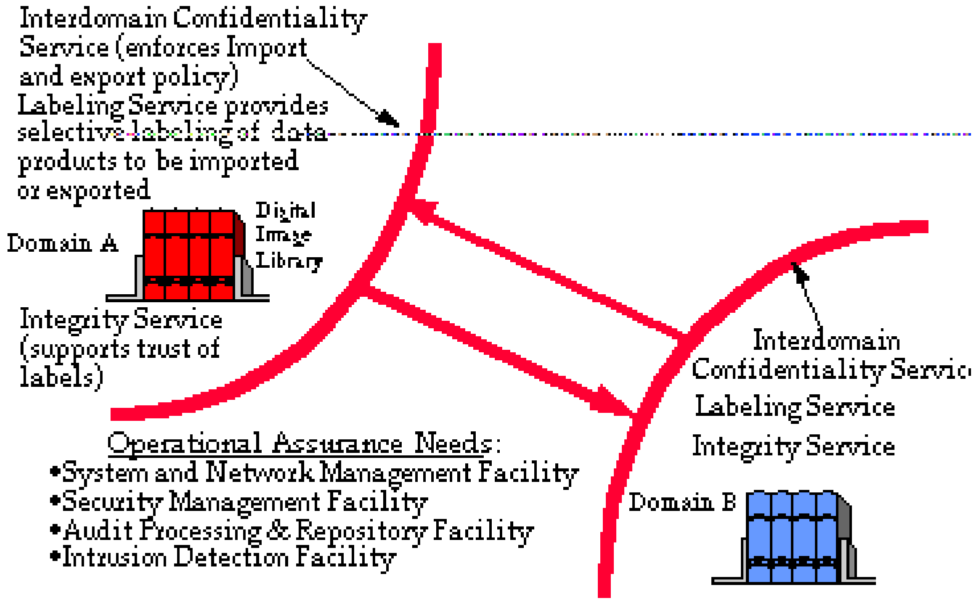
Interdomain Confidentiality Service (enforces Import and export policy)
Labeling Service provides selective labeling of data products to be imported or exported
Digital Image Library
Domain A
Integrity Service (supports trust of labels)
Operational Assurance Needs:
•System and Network Management Facility
•Security Management Facility
•Audit Processing & Repository Facility
•Intrusion Detection Facility
Interdomain Confidentiality Servic
Labeling Service
Integrity Service
Domain B

Figure 4-6 Interdomain Confidentiality Service

In certain domains, security accreditors may find that the level of assurance provided is not sufficient to protect the interdomain confidentiality service and the other services from interference from other internal processes. In that case, a separate domain with the standard interdomain confidentiality and related services implemented on a separate platform may provide additional assurance. This separate platform configuration can be thought of as a distributed computing system security guard.

4.5.4 Summary of Recommended Security Services

The following table defines the security services that may be required by security policy domains operating in the system high (e.g., system-high mode policy domain), compartmented, and multilevel mode of operation.

Table 4-2 Recommended Security Services

Legend Priority of service for the security modes. For example, "P1, CM, MLS" means P1 for the service used in a system that will operate in CM and MLS modes. The service is optional for SH mode. P1 - The standard interface must be defined. Service must be implemented and be used appropriately by the middleware or an application (policy must be enforced) for the system to pass certification and accreditation (C&A). P2 - The standard interface must be defined. Service implementation and use is required. Cost - benefit analysis and negotiation with C&A authority may wave requirement for implementation and/or use until service is feasible. P3 - A system can be accredited in any security mode without this service. However, the service may be desired by the appropriate mission application to help the user/analyst make security related decisions (e.g., what is the classification of my data fused intelligence report?) The standard interface should be defined, so that vendors may access marketability of implementation.	SH - Required by System-High Mode Policy Domains CM- Required by Compartmented Mode Policy Domains MLS - Required by Multilevel Mode Policy Domains Ap - Desired, Application in any policy domain may employ to aid user Y - Middleware provides concept or service D - Directly supported, but not provided by middleware DS - Same as D, middleware may supply in future C - Consistent with middleware, but no plans for middleware to supply IF - Middleware provides protection only at interface level of granularity for services or objects (does not protect individual data structures or instances of objects) APC - Applications must provide security (for example, access control) NM - Not provided by middleware, should be implemented in another part of the infrastructure

USIGS Standard Security Service or Functionality	Description	Priority and Security Modes
1. Accountability	Services provided to guarantee that all individual users are accountable for their actions while interacting with USIGS.	
A. Identification and Authentication	Verifies information from user or other entity (e.g., object) to confirm identity.	P1, SH, CM, MLS, Ap
I&A Individual User to Trusted Login Process	Platform operating system uses trusted process (for example, login shell) to get userid and password from user and provides this authentication data to ORB or process.	P1, SH, CM, MLS
I&A User/Client to/from Server Application	Application uses service to authenticate users/clients. Clients may use service to authenticate application. User's credentials must be delegated to objects representing the user and vice versa.	P2, Ap
I&A Process to Process or Object to Object	Object or application uses service to authenticate target object or application.	P2, SH, CM, MLS, Ap
I&A Process/Object to/from Device	Object or application uses service to authenticate target device (and vice versa) (for example, object to printer).	P1, CM, MLS, Ap
I&A Device to Device	Device (for example, workstation platform uses service to authenticate system management station and vice versa.	P2, CM, MLS
B. Auditing	This service is concerned with the generation, storage, processing and management of the accountability information.	P1, SH, CM, MLS, Ap
Audit Event Server	Objects, applications, security functions, and other system software use service to generate security related (audit) events and alarms, server collects events, sends active events to audit manager repository, sends alarm events to security alarm manager. Event collection by the server is managed (i.e., events turned on/off) by audit event manager service (see security management services). Audit event server may subscribe to general system event server.	P1, SH, CM, MLS, Ap
Generate Audit Record or Security Alarm	Gives the capability to an application to generate an audit record or security alarm.	P1, SH, CM, MLS, Ap
Notify Security Alarm Manager	Audit Event Server sends priority alert message to security management server when: •Number of events of a certain type, for particular user, reach certain threshold. •Receives a security alarm.	P1, SH, CM, MLS, Ap

Audit Repository Processing Facility	Stores audit records and provides capabilities to analyze stored records.	SH, CM, MLS, AP

Table 4-2 Recommended Security Services (continued)

Intrusion Detection Facility	Service to provide real-time detection of anomalous behavior by comparison of usage profiles with active sessions. This service required mainly at interdomain gateway.	P1, SH, CM, MLS
2. Intradomain Confidentiality	Services provided to prevent unauthorized disclosure of information or unauthorized disclosure of information within a policy domain.	
A. Labeling	Labels that indicate the sensitivity of the associated object or data structure. Required to support access control services. Must use the integrity service so the generated label can be trusted.	P1-P3, SH, CM, MLS
Get Sensitivity Label for Label-Based Access Control	Labeling required to support label-based access control (for example, rule-based access control), requires use of integrity service so label can be trusted. Label used at interface, instance, and data structure levels of granularity.	P1, CM, MLS
Get Domain Export Label	Sensitivity label required to be associated with all data structures exported out of policy domain. For system-high mode, it is optional to label non-imagery data instances.	P1, SH, CM, MLS
Get Domain Import Label	Sensitivity label required to be associated with all data structures imported into policy domain. For system-high mode, it is optional to label non-imagery data instances.	P1, SH, CM, MLS
Get Device Label	Device level granularity labeling.	P1, CM, MLS
Get Interface Label	Object or server interface granularity labeling.	P1, CM, MLS
Fine Grained Data Labeling	Granular labeling within object's or server's encapsulated data structures (for example, database fields).	P3, Ap
Imagery Sensitivity Labeling	Used in all security policy domains to provide trusted labeling of imagery. Integrity service is used to provide sufficient integrity (for example, cryptographic checksum, digital signature). This label is required to support the interdomain confidentiality service.	P1, SH, CM, MLS
B. Informational Labeling	Labeling for informational purposes only, not used by access control, requires use of integrity service if label needs to be trusted.	P3, Ap
Get Informational Label	Determine value of informational label.	P3, Ap
Floating Informational Labels	Information labels dynamically change to represent true sensitivity as data of different sensitivities flows into or out of object or application.	P3, Ap
C. Relabeling Support	Trusted support for new labeling or changing existing labels associated with downgrading an archive of image metadata.	
Filtering and Aging for Downgrading or Upgrading	Selecting out for example, SCI imagery metadata from for example, Secret metadata based on approved search criteria (not necessarily label-based), then associating a trusted sensitivity label with the metadata (sensitivity or information labeling supported).	P2, SH, CM, MLS, Ap
Re-label Archive After Policy Change	Provides for trusted new labeling or relabeling of data based on policy change (for example, places new labels on all data in an archive based on new policy criteria that allows downgrade of imagery data from SCI to Secret) (sensitivity or information labeling supported).	P2, SH, CM, MLS, Ap
D. Interface Access Control	The objective is that there be functionality and associated databases necessary to support rule-based and identity-based access controls at the interface level of granularity. Can control access to applications and services (for example, Imagery Access Server).	P1, CM, MLS
Rule-Based Access Control (RBAC)	Rule-based access controls use formal and mandatory rules to mediate access. The rules reflect mandatory access control policy.	P1, CM, MLS
Update RBAC database	Add/delete/modify/etc. access rules. A database of rules must be maintained and managed by the security management services. Policy changes can be made by changing the rules.	P1, CM, MLS
Get Rule-Based Attributes	Get object's RBAC attribute(s), for example, sensitivity label. This interface can be called for both client and server objects	P1, CM, MLS
RBAC Mediation	Interprets and applies appropriate rules, given type of access and RBAC attributes, returns access granted or denied.	P1, CM, MLS

Identity-Based Access Controls (IBAC)	Identity-based access controls are to be used when access to an object or service interface is to be restricted based on individual identity or membership in a group. Supports setting up long or short term COIs. Access types are extensible.	P1, SH, CM, MLS

Table 4-2 Recommended Security Services (continued)

Update IBAC Attributes	Create, modify, append attributes that indicate who can access object or services (for example, update object's or service's access control list or User-Group-World permissions).	P1, SH, CM, MLS
Get Requester IBAC Context	Gets the requesting object's or service's IBAC identity context, for example, UserID. May involve determining if delegation has occurred.	P1, SH, CM, MLS
Get Target IBAC Attributes	Gets the target object's or service's IBAC attributes (for example, access control list or User-Group-World permissions).	P1, SH, CM, MLS
IBAC Mediation	IBAC mediation service function compares user identity information, such as UserID, with IBAC attributes associated with the requested object or service, that indicates who may invoke the object or service. Calling object or service specifies if any delegation is to be applied during mediation.	P1, SH, CM, MLS
Need to define access controls that mediate access at data structure level of granularity. Recommend further study.	Data structures are encapsulated within objects or servers (for example, image server application object controls all image data structures). Access control to data structures is allocated to applications. Applications should reuse a standard set of interfaces/functionality to implement this security service.	P1, SH, CM, MLS, Ap
Delegation with User Specific Auditing	Used by an object to empower another object to act on its behalf, CORBA currently only provides for impersonation. USIGS requires that audit trails must identify the user who initiated the original request	P1, SH, CM, MLS
3. Other services		
A. End-to-End Transaction Compromise Protection	Used to provide transactions with end-to-end protection of data from compromise. This protection may be necessary for privacy reasons (for example, time sheet or medical record). Applications may use this service to implement end-to-end protection.	P1, SH, CM, MLS
B. Communications Data Compromise Protection	It is an objective that a service be provided to protect t system message traffic from compromise as it is transmitted by the TCP/IP protocol mechanisms of the underlying communications infrastructure. This may be implemented by physical or cryptographic means.	P1, SH, CM, MLS
C. Selective Routing	It is an objective that a service be provided to selectively route system message traffic as it is transmitted by the TCP/IP protocol mechanisms of the underlying communications infrastructure.	P2, SH, CM, MLS
4. Interdomain Confidentiality Services	A major service needed by USIGS is interdomain confidentiality. This service must enforce a domain import/export policy that specifies what services and data instances can be accessed by dynamic request from/to an external domain. The service can be thought of as an interdomain import/export access control service.	P1, SH, CM, MLS
A. SIIOP Firewall Proxy Service	This service provides a protocol filtering mechanism that intercepts all SIIOP message and data instance traffic and passes it to the appropriate ORB or security gateway application	
B. Domain Validation	This function can be used to implement the mutually suspicious domains concept. Domain credentials are exchanged, and the domains are mutually authenticated. Individual domain administrators can use this service to enforce any domain-to domain security handshaking they deem appropriate.	P1, SH, CM, MLS
A. Domain Proxy Service	If out going object requests (e.g., digital image server access request) are to be mediated (i.e., can a user or associated object or process make a request to a particular server outside the local domain), then all external servers or objects accessible from the local domain need to be proxied in the local domain.	P1, SH, CM, MLS
B. Interdomain Interceptor	An interceptor (i.e., software that lives in the ORB, that mediates incoming/outgoing message or data traffic by calling a security gateway server application or directly using interdomain validation functionality.	P1, SH, CM, MLS

D. Interface Validation (Verify Service Request)	Validates that an object can invoke another object or service. The intradomain interface access control service can be reused.	P1, SH, CM, MLS
F. Imagery Validation	Used to verify that imagery data structure instance is allowed to be transferred to the requesting domain. Verification involves at least checking the label, but may also validate the data structure.	P1, SH, CM, MLS

Table 4-2 Recommended Security Services (continued)

E. Data Validation	Used to validate pre-defined data structures other than imagery. Checks the actual data that is being transferred between domains for conformance to restrictions concerning format and content, for example, check for restricted codewords. This service is provided for use in domains that allow other that imagery data structures to be imported/exported.	P1, SH, CM, MLS
G. Integrity Validation	Verifies that messages, objects, data structures, attributes, formats, and their content have not been subject to unauthorized modification.. This service may reuse the intradomain integrity service.	P1, SH, CM, MLS
H. Get Imagery Sensitivity Label	Used to get the label associated with the image being sent to the other domain. This operation reuses the integrity service to provide and verify the integrity of the label. This function can reused the intradomain service counterpart if it exist.	P1, SH, CM, MLS
K. Mediate Interdomain Imagery Import/Export Access Policy	Interprets and applies appropriate interdomain imagery import/export access rules, returns access granted or denied.	P1, SH, CM, MLS
L. Update Interdomain Import/Export Policy Rules	Add/delete/modify/etc. interdomain imagery label-based access rules. A database of rules must be maintained and managed by the security management services. Policy changes can be made by changing the rules.	P1, SH, CM, MLS
5. Integrity	Used to verify that messages, objects, data structures, labels, attributes, formats, protocol data units, and their content have not been subject to unauthorized modification.	
A. Integrity Shield	Used to convert data into integrity protected data.	P1, SH, CM, MLS
B. Integrity UnShield	Used to convert integrity protected data into the original data.	P1, SH, CM, MLS
C. Integrity Validate	Used to check integrity protected data to detect loss of integrity.	P1, SH, CM, MLS
5. Non-Repudiation	There must be a USIGS non-repudiation service that provides generation of evidence of actions and later verification of this evidence, to prove that the action has occurred.	
A. Evidence generation and verification	Generation and verification of evidence of an action.	P2, SH, CM, MLS
B. Evidence Storage and Retrieval	Storage and retrieval of evidence.	P2, SH, CM, MLS
C. Delivery Authority	Third party authority.	P2, SH, CM, MLS
6. Operational Assurance		
A. Security of Management Functions	Access control services are used to provide protection. Access is based on identity-based roles.	P1, SH, CM, MLS
B. System and Network Management	System and network management is concerned with the management of system and network services, mechanisms that implement the services, as well as the associated information data structures concerning the operation of the services and mechanisms. This area needs further study, but is not within the scope of this study.	P1, SH, CM, MLS
C. Security Management	Centralized management of distributed security mechanisms . This area needs further study.	P1, SH, CM, MLS

Appendix I
DEFINITION OF TERMS

Adoption - The acceptance and approval of a Government or non-Government standard.

Application Program Interface (API) – The collection of available function calls (or other input/output mechanisms) that enable other systems to obtain services from, exchange data with, or otherwise interact with an application program.

Audience – The types of the intended consumers (callers) of an interface; similar in concept to the client side in the client/server software model.

Bearer – The object type that presents an interface; similar in concept to the server side in the client/server software model.

Certification - A statement attesting to the fact that a software product has been verified to conform to Facility specifications.

Common Facilities – A collection of higher-level services that are broadly applicable across many different application domains.

Common Imagery Interoperability Facilities (CIIF) – The collection of interface structures and services that will be rigorously standardized (using ISO's Interface Definition Language - IDL) to achieve effective interoperability among the digital elements that comprise the USIGS Technical Architecture. The CIIF can be viewed as comprising the public API for the USIGS. Also referred to as *Facilities*.

Common Imagery Interoperability Working Group (CIIWG) – The Government-sponsored, open-to-the-public consortium that is chartered to oversee the definition, design, and development of the CIIF. The goal of the CIIWG is to ensure that the concerns of all interested parties (including commercial entities, the standards community, and other governmental organizations) are considered during the development of the CIIF.

Object Service Interfaces - A collection of fundamental services (interfaces and objects) that provide basic functions for using and implementing distributed software objects. A variety of distributed computing standards are currently evolving such as CORBA, OLE, and DCE.

Configuration Control - The systematic proposal, justification, evaluation, coordination, approval or disapproval of proposed changes, and the implementation of all approved changes, in the configuration item after establishment of its baseline.

Configuration Control Board (CCB) - A board composed of technical and administrative representatives who recommend approval or disapproval of proposed engineering changes to a CI's current approved configuration documentation. The board also recommends approval or disapproval of proposed waivers and deviations from a CI’s current approved configuration documentation.

Configuration Management (CM) - As applied to configuration items, a discipline applying technical and administrative direction and surveillance over the life cycle of items to:

a. Identify and document the functional and physical characteristics of configuration items.

b. Control changes to configuration items and their related documentation.

c. Record and report information needed to manage configuration items effectively, including the status of proposed changes and implementation status of approved changes.

d. Audit configuration items to verify conformance to specifications, drawings, interface control documents, and other contract requirements. (MIL-STD-973)

Conformance - adherence to a Facility's specifications

Consensus - Consensus by the CIIWG is achieved by simple majority vote.

Coordination - The process of having standardization documents reviewed and commented on by Government and private sector organizations.

Distributed Computing Infrastructure – Enables objects to make/receive requests/responses in a distributed computing environment. A variety of distributed computing standards are currently evolving, such as CORBA, OLE, and DCE.

Department of Defense Index of Specifications and Standards (DODISS) - A publication that lists Federal and military specifications and standards, guide specifications, military handbooks and bulletins, commercial item descriptions, adopted non-government standards, and other related standardization documents used by the DoD.

Executive Agent - CIIWG member organization tasked to develop assigned facilities.

Facility - See **Common Imagery Interoperability Facility**

Facility Maintenance - see **Configuration Control**

Geospatial Information Services – Interfaces to certain high-value services or capabilities that are specific to the imagery industry or application domain.

Implementation - Addresses operational assessment, CM, acquisition support, and enforcement.

Inheritance – An object-oriented programming concept in which crucial features of one software artifact are automatically passed to (inherited by) a subordinate software artifact. The Object Management Group's *Interface Definition Language* (IDL) includes support for multiple inheritance, in which a software interface object can inherit structural features from several other such objects at once.

Interface Definition Language (IDL) – A formal language (similar in appearance to a C++ header file) that is used to define the interfaces between interoperable software objects; can be directly compiled into any of several common programming languages, using standard language mappings, to automatically set up the mechanisms needed to pass service requests across the network in a distributed software architecture.

Object Management Group (OMG) – A consortium of over 500 commercial and governmental organizations that oversees the development of certain object-oriented standards, methods, and technologies including IDL.

Reference Model - A reference model is intended to establish a framework for the development of consistent standards or specifications. A reference model is based on a small number of unifying concepts and may be used as a basis for education and explaining standards to a non-specialist. (A reference model should be compatible with other, existing reference models to the extent practical.)

Specification - A Facility's IDL code.

Standard - Refers to a procedure, protocol, and development of specifications within any research and development technology area that will promote interoperability and compatibility of two or more components within the United States Imagery and Geospatial Information System (USIGS). The Imagery and Research Development Council will recognize certain technical standards, those that are evolving or the absence of needed standards in making sure that research and development efforts will be successfully implemented as operational entities in the USIGS.

Testing - Test procedures and tools are used to verify that a sample implementation conforms to the Facility specifications.

United States Imagery System (USIS) – The people, systems, and procedures (both current and future) that support the end-to-end production of imagery and imagery-related intelligence products. The USIS Technical Architecture, which focuses on defining both the near-term and long-term architecture of imagery-related *systems*, has been partitioned into eight digital elements:

- Management Element
- Digital Collection Element
- Digital Processing Element
- Dissemination Element
- Library Element
- Digital Exploitation Element
- Site Infrastructure Element
- Global Communications Element.

Mission Area Applications – Software objects specific to a the USIGS. For example, the six application-layer digital elements in the USIGS Technical Architecture (Management, Digital Collection, Digital Processing, Dissemination, Library, and Digital Exploitation) can be viewed as consisting (principally) of USIGS Application objects.

Appendix II
ACRONYMS

A^{3}I - Accelerated Architecture Acquisition Initiative
ANSI - American National Standards Institute
API - Application Program Interface
ARD - Architecture Requirements Document
ATR - Automatic Target Recognition

CAF - Catalog Access Facility
CCB - Configuration Control Board
CI - Configuration Item
CIGSS - Common Imagery Ground / Surface System
CIIF - Common Imagery Interoperability Facility
CIIWG - Common Imagery Interoperability Working Group
CIL - Command Image Library
CIO - Central Imagery Office
CM - Configuration Management
COAX - Coaxial
COE - Common Operating Environment
COM - Component Object Model (Microsoft)
CONOPS - Concepts of Operation
CORBA - Common Object Request Broker Architecture

DARO - Defense Airborne Reconnaissance Office
DCE - Distributed Computing Environment
DCI - Director Central Intelligence
DIA - Defense Intelligence Agency
DII COE - Defense Information Infrastructure Common Operating Environment
DISA CFS - Defense Information Systems Agency Center for Standards
DMA - Defense Mapping Agency
DOD - Department of Defense
DODISS - Department of Defense Index of Specifications and Standards
DSP - Defense Standardization Program

EEI - External Environment Interface
ESIOP - Environment Specific Inter-ORB Protocol

FDDI - Fiber Distributed Data Interface
5D - Demand Driven Direct Digital Dissemination
FTP - File Transfer Protocol

GCCS - Global Command & Control System
GIOP - Generic Inter-ORB Protocol

HAE UAV - High Altitude Endurance Unmanned Aerial Vehicle
HTTP - Hyper Text Transfer Protocol

IAB - Internet Activities Board
IAF - Image Access Facility
IC - Intelligence Community
ICCB - Imagery Configuration Control Board
ICD - Interface Control Document
IDEX - Imagery Data Exploitation System
IDL - Interface Definition Language
IIOP - Internet Inter-ORB Protocol
IMS - Interoperable Map Software
IP - Internet Protocol
IPA - Image Product Archive
IPL - Image Product Library
ISB - Intelligence Systems Board
ISDN - Integrated Services Digital Network
ISMC - Imagery Standards Management Committee
ISO - International Standards Organization
ISS - Intelligence Systems Secretariat

JIEO - Joint Interoperability and Engineering Organization
JITC - Joint Interoperability Testing Command
JMTK - Joint Mapping Tool Kit
JRD - Joint Requirements Document
JTA - Joint Technical Architecture

MATRIX - Modular Automated Target Recognition for Interactive Exploitation
MIDB - Modernized Intelligence Database
MIL-STD - Military Standard
MINT - Multi-source Intelligence Tools

NEL - National Exploitation Lab
NIL - National Image Library
NITFS - National Imagery Transmission Format Standards
NPIC - National Photographic Interpretation Center
NTB - NITFS Technical Board

OASD - Office of the Assistant Secretary of Defense
OGC - Open GIS Consortium
OGIS - Open Geodata Interoperability Specification™
OLE - Object Linking & Embedding (Microsoft)
OMA - Object Management Architecture
OMG - Object Management Group
ORB - Object Request Broker
OS - Operating System
OSF - Open Systems Foundation
OSI - Open Systems Interconnect

PIKS - Programmer’s Imaging Kernel System
POSIX - Portable Operating System for Information Exchange
PPP - Point-to-Point Protocol

RFC - Request for Change
RPC - Remote Procedure Call

SCC - Standards Coordinating Committee
SECDEF - Secretary of Defense
SMTP - Simple Mail Transfer Protocol
SPIA - Standards Profile for Imagery Access

TAFIM - Technical Architecture Framework for Information Management
TCP - Transmission Control Protocol

UDP - User Datagram Model
USIS - United States Imagery System
USIS S&G - United States Imagery System Standards & Guidelines
USIS TAR - United States Imagery System Technical Architecture Requirements

VWG - Video Working Group

WG - Working Group
W3C - World Wide Web Consortium